Computational Elements for Strapdown Systems

Paul G. Savage
Strapdown Associates, Inc.
Maple Plain, Minnesota 55359 USA

ABSTRACT

This paper provides an overview of the primary strapdown inertial system computational elements and their interrelationship. Using an aircraft type strapdown inertial navigation system as a representative example, the paper provides differential equations for attitude, velocity, position determination, associated integral solution functions, and representative algorithms for system computer implementation. For the inertial sensor errors, angular rate sensor and accelerometer analytical models are presented including associated compensation algorithms for correction in the system computer. Sensor compensation techniques are discussed for coning, sculling, scrolling computation algorithms and for accelerometer output adjustment for physical size effect separation and anisoinertia error. Navigation error parameters are described and related to errors in the system computed attitude, velocity, position solutions. Differential equations for the navigation error parameters are presented showing error parameter propagation in response to residual inertial sensor errors (following sensor compensation) and to errors in the gravity model used in the system computer.

COORDINATE FRAMES

As used in this paper, a coordinate frame is an analytical abstraction defined by three mutually perpendicular unit vectors. A coordinate frame 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 position of each coordinate frame's origin is arbitrary. The principal coordinate frames utilized are the following:

B Frame = "Body" coordinate frame parallel to strapdown inertial sensor axes.

N Frame = "Navigation" coordinate frame having Z axis parallel to the upward vertical at the local position location. A "wander azimuth" N Frame has the horizontal X, Y axes rotating relative to non-rotating inertial space at the local vertical component of earth's rate about the Z axis. A "free azimuth" N Frame would have zero inertial rotation rate of the X, Y axes around the Z axis. A "geographic" N Frame would have the X, Y axes rotated around Z to maintain the Y axis parallel to local true north.

E Frame = "Earth" referenced coordinate frame with fixed angular geometry relative to the rotating earth.

I Frame = "Inertial" non-rotating coordinate frame.

NOTATION

$\underline{V}$ = Vector without specific coordinate frame designation. A vector is a parameter that has length and direction. Vectors used in the paper are classified as "free vectors", hence, have no preferred location in coordinate frames in which they are analytically described.

$\underline{V}^A$ = Column matrix with elements equal to the projection of $\underline{V}$ on Coordinate Frame A axes. The projection of $\underline{V}$ on each Frame A axis equals the dot product of $\underline{V}$ with the coordinate Frame A axis unit vector.

$\left(\underline{V}^A \times\right)$ = Skew symmetric (or cross-product) form of $\underline{V}^A$ represented by the square matrix $\begin{bmatrix} 0 & -V_{ZA} & V_{YA} \\ V_{ZA} & 0 & -V_{XA} \\ -V_{YA} & V_{XA} & 0 \end{bmatrix}$ in which V_{XA}, V_{YA}, V_{ZA} are the components of $\underline{V}^A$. The matrix product of $\left(\underline{V}^A \times\right)$ with another A Frame vector equals the cross-product of $\underline{V}^A$ with the vector in the A Frame.

$C_{A_2}^{A_1}$ = Direction cosine matrix that transforms a vector from its Coordinate Frame A_2 projection form to its Coordinate Frame A_1 projection form.

$\underline{\omega}_{A_1 A_2}$ = Angular rate of Coordinate Frame A_2 relative to Coordinate Frame A_1. When A_1 is non-rotating, $\underline{\omega}_{A_1 A_2}$ is the angular rate that would be measured by angular rate sensors mounted on Frame A_2.

$\dot{(\ )} = \frac{d(\ )}{dt}$ = Derivative with respect to time.

t = Time.

1. INTRODUCTION

The primary computational elements in a strapdown inertial navigation system (INS) consist of integration operations for calculating attitude, velocity and position navigation parameters using strapdown angular rate and specific force acceleration for input. The computational form of these operations originate from two basic sources: time rate differential equations for the navigation parameters and analytical error models describing the error characteristics of the strapdown inertial angular rate sensors and accelerometers providing the angular rate and specific force acceleration measurement data. The latter is the source for compensation algorithms used in the system computer to correct predictable errors in the inertial sensor outputs. The former is the source for digital integration algorithms resident in system software for computing the navigation parameters. Both are the source for error propagation equations used to describe the behavior of navigation parameter errors in the presence of residual sensor errors remaining after compensation.

This paper provides examples of each of the aforementioned computational elements and their interrelationship. For the digital integration algorithms, the examples are selected to emphasize a structural goal of being based (to the greatest extent possible) on closed-form analytically exact integral solutions to the navigation parameter time rate differential equations. Such a structure significantly simplifies the integration algorithm software validation process based on a comparison with closed-form exact solution dynamic model simulators designed to thoroughly exercise the exact solution algorithms under test (Reference 26). For properly derived and programmed algorithms, the comparison will yield identically zero difference, thereby providing a clear unambiguous algorithm software validation. Once validated, such algorithms can be used as a generic set suitable for all strapdown inertial applications. Associated algorithm documentation is also simplified because algorithm derivations are classical analytical formulations and explanations/numerical-error-analysis justification for application dependent approximations are not required because there are none. Modern day strapdown system computer technology (high throughput, long floating point word-length) allows the general use of such exact solution algorithms without penalty. Similarly, the sensor compensation algorithms shown in the paper are a generic set based on the exact inverse of classical sensor error models without first order approximations (as has been commonly used in the past to save on computer throughput).

The form of the navigation error propagation equations are based on analytical definitions of the attitude, velocity, position error parameters. Several choices are possible. Two of the most common sets are illustrated in the paper and equivalencies between the two described. An example of the error propagation equations based on one of the sets is provided.

This paper is an updated version of Reference 22. Reference 22 is a condensed summary of material originally published in the two volume textbook *Strapdown Analytics* (Ref. 20), the second edition of which has been recently published (Reference 25). *Strapdown Analytics* provides a broad detailed exposition of the analytical aspects of strapdown inertial navigation technology. This version of the Reference 22 paper also incorporates new material from the recently published paper *A Unified Mathematical Framework For Strapdown Algorithm Design* (Reference 23) - also provided in Section 19.1 of the second edition of *Strapdown Analytics* (Reference 25). Equations in this paper (as in Reference 22) are presented without proof. Their derivations are provided in Reference 20 (or 25) and in Reference 23 as delineated throughout the paper (by Reference 20 or 25 section number and by Reference 23 equation number). Documents delineated in the paper's References listing that are not cited in the body of the paper are those cited in Reference 20 (or 25) that are specifically related to the paper's subject matter.

2. REPRESENTATIVE STRAPDOWN INERTIAL NAVIGATION DIFFERENTIAL EQUATIONS

This section describes a typical set of basic attitude/velocity/position integration and acceleration transformation operations performed in a strapdown INS. The integration operations are described in the form of continuous differential equations that when integrated in the classical analytical continuous sense, provide the attitude, velocity and position data generated digitally in the strapdown system computer. The algorithms described in Section 4 are designed to achieve the same numerical result by digital integration as the continuous integration of the differential equations presented in this section.

2.1 Attitude

For a terrestrial (earth) based inertial navigation system (e.g., for aircraft), sensor assembly angular attitude orientation is usually described as an “attitude direction cosine matrix” (or attitude quaternion) relating sensor assembly axes (the “body” or B Frame) to locally level attitude reference coordinates (N Frame). Attitude determination consists of integrating the associated time rate differential equations for the selected attitude parameters. For an attitude reference formulation based on direction cosines the attitude time rate differential equations are given by (Ref. 20 (or 25) Sects. 4.1 and 4.1.1):

$$\dot{C}_B^N = C_B^N \left(\underline{\omega}_{IB}^B \times\right) - \left(\underline{\omega}_{IN}^N \times\right) C_B^N$$

$$\underline{\omega}_{IE}^N = \left(C_N^E\right)^T \underline{\omega}_{IE}^E \qquad \underline{\omega}_{EN}^N \equiv \underline{\rho}^N = F_C^N \left(\underline{u}_{Up}^N \times \underline{v}^N\right) + \rho_{ZN}\, \underline{u}_{ZN}^N \tag{1}$$

$$\underline{\omega}_{IN}^N = \underline{\omega}_{IE}^N + \underline{\omega}_{EN}^N$$

where

$\underline{\rho}^N$ = Conventional notation for $\underline{\omega}_{EN}^N$, also known as “transport rate”, and analytically defined as the angular rate of Frame N relative to Frame E.

ρ_{ZN} = Vertical component of $\underline{\rho}^N$. For a "wander azimuth" N Frame, ρ_{ZN} is zero. For a "free azimuth" N Frame, ρ_{ZN} is the downward vertical component of earth's inertial angular rate.

F_C^N = Curvature matrix in the N Frame that is a function of position location over the earth.

$\underline{v}$ = Velocity (rate of change of position) relative to the earth.

$\underline{u}_{Up}$ = Unit vector upward at the current position location (parallel to the N Frame Z axis).

The equivalent quaternion formulation (Ref. 20 (or 25) Sect. 4.1) is as follows:

$$\dot{q}^N_B = \frac{1}{2}\, q^N_B\, \omega^B_{IB} - \frac{1}{2}\, \omega^N_{IN}\, q^N_B \tag{2}$$

where

q^N_B = Attitude quaternion relating coordinate Frames B and N.

$\omega^B_{IB}, \omega^N_{IN}$ = Quaternions with vector components equal to $\underline{\omega}^B_{IB}, \underline{\omega}^N_{IN}$ and zero for the scalar components.

The C^E_N matrix in Equations (1) defines the system angular position location in earth reference coordinates, hence, is sometimes denoted as the "position" direction cosine matrix (or the equivalent position quaternion). The C^E_N matrix is calculated by integrating its differential equation (described in Section 2.3) using $\underline{\omega}^N_{IN}$ (N Frame "platform" rotation rate) as input. For earth's zero altitude surface reference modeled as an ellipsoid of revolution around earth's rotation axis (i.e., the conventional approach), Reference 20 (or 25) Sections 5.2.4 and 5.3 develop the following exact expression for the F^N_C curvature matrix in Equations (1) based on an E Frame definition having Y axis parallel to earth's axis of rotation:

$$F^N_C = \begin{bmatrix} F_{C_{11}} & F_{C_{12}} & 0 \\ F_{C_{21}} & F_{C_{22}} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} F_{C_{11}} &= \frac{1}{r_l}\left(1 + D_{21}^2\, f_{eh}\right) \qquad & F_{C_{12}} &= \frac{1}{r_l} D_{21}\, D_{22}\, f_{eh} \\ F_{C_{21}} &= \frac{1}{r_l} D_{21}\, D_{22}\, f_{eh} \qquad & F_{C_{22}} &= \frac{1}{r_l}\left(1 + D_{22}^2\, f_{eh}\right) \end{aligned} \tag{3}$$

$$r_l = R_0 \frac{(1 - e)^2}{\left\{1 + D_{23}^2 \left[(1 - e)^2 - 1\right]\right\}^{3/2}} + h$$

$$f_{eh} \equiv \frac{(1 - e)^2 - 1}{\left(1 + D_{23}^2 \left[(1 - e)^2 - 1\right]\right)\left(1 + \frac{h}{R_0}\sqrt{1 + D_{23}^2 \left[(1 - e)^2 - 1\right]}\right)}$$

where

D_{ij} = Element in row i column j of C^E_N.

e = Ellipticity of earth's reference surface ellipsoid.

R_0 = Earth's equatorial radius.

r_l = Local radius of curvature at altitude in the North/South (latitude change) direction.

h = Altitude from earth's reference surface ellipsoid to the current position location (positive above the earth's surface).

2.2 Velocity

The velocity data in an inertial navigation system is typically computed as an integration of velocity rate described in the navigation N Frame. The velocity of interest is usually defined as the time rate of change of position relative to the earth in a coordinate frame that rotates at earth's rotation rates (i.e., the E Frame):

$$\underline{v}^E \equiv \dot{\underline{R}}^E \tag{4}$$

where

$\underline{R}$ = Position vector from earth's center to the current position location.

In the N Frame, the velocity is then:

$$\underline{v}^N = C_E^N \underline{v}^E \tag{5}$$

Based on this definition, the time rate differential equation for velocity is (Ref. 20 (or 25) Sect. 4.3):

$$\dot{\underline{v}}^N = C_B^N \underline{a}_{SF}^B + \underline{g}^N - \underline{\omega}_{IE}^N \times \left(\underline{\omega}_{IE}^N \times \underline{R}^N\right) - \left(\underline{\omega}_{IN}^N + \underline{\omega}_{IE}^N\right) \times \underline{v}^N \tag{6}$$

where

$\underline{a}_{SF}$ = Specific force acceleration defined as the instantaneous time rate of change of velocity imparted to a body relative to the velocity it would have sustained without disturbances in local gravitational vacuum space. Sometimes defined as total velocity change rate minus gravity. Accelerometers measure $\underline{a}_{SF}$.

$\underline{g}$ = Mass attraction gravity at the current position location minus mass attraction gravity at the center of the earth. Sometimes denoted as "gravitation" (Ref. 2 Sect. 4.4).

For the quaternion attitude formulation approach in Section 2.1, the $C_B^N \underline{a}_{SF}^B$ term in Equation (6) would be replaced by the vector part of the quaternion product $q_B^N \underline{a}_{SF}^B q_B^N{}^*$ in which $q_B^N{}^*$ is the conjugate of q_B^N and a_{SF}^B is the quaternion with $\underline{a}_{SF}^B$ for its vector component and zero for its scalar component. Alternatively, once q_B^N is calculated by integrating Equation (2), it can be converted to the equivalent C_B^N direction cosine matrix (Ref. 20 (or 25) Sect. 7.1.2.4) which is then directly compatible with Equation (6) as shown.

Reference 20 (or 25) Section 5.4.1 shows how $\underline{g}^N - \underline{\omega}_{IE}^N \times \left(\underline{\omega}_{IE}^N \times \underline{R}^N\right)$ in Equation (6) can be calculated without singularities based on a classical gravity model defined in the E Frame (Ref. 2 Sect. 4.4 and Ref. 3). The latter references model gravity on and above earth's zero altitude surface. Reference 20 (25) Section 5.4 extends the model for negative altitudes (i.e., below earth's surface).

2.3 Position

Position relative to the earth is often described by altitude above the earth and the angular orientation of the current local vertical direction in earth coordinates (the E Frame). The angular position parameters are commonly represented by latitude and longitude, however, to avoid mathematical singularities, the angular position parameters are frequently represented in the form of the N to E position direction cosine matrix (or the equivalent quaternion). The time rate differential equations for the position direction cosine matrix and altitude are as follows (Ref. 20 (or 25) Sects. 4.4.1.1 and 4.4.1.2):

$$\dot{C}_N^E = C_N^E \left(\underline{\rho}^N \times\right) \qquad \dot{h} = \underline{u}_{Up}^N \cdot \underline{v}^N \tag{7}$$

2.4 Attitude, Velocity, Position Output Conversion

An advantage for using C_B^N, C_N^E (or their quaternion equivalents), $\underline{v}^N$, and h as the basic navigation parameters calculated by integration is that the associated differential equations have no singularities for all

INS attitude orientations and position locations. Once calculated, they can be output from the INS directly and/or converted into other formats for output (e.g., roll, pitch, heading attitude; north, east, vertical velocity; latitude, longitude, altitude position - Ref. 20 (or 25) Sects. 4.1.2, 4.3.1, and 4.4.2.1).

3. Integral Solutions For The Navigation Parameters

The digital integration algorithms resident in the strapdown system computer are based on integrated forms of the Section 2 navigation parameter differential equations over a digital integration update cycle. For modern day algorithms, the integrated form is structured into two operations; 1. Basic digital updating operations used to increment the attitude/velocity/position parameters over each update cycle, and 2. High speed integration operations that account for high frequency angular-rate/acceleration inputs between each update cycle (coning effects in attitude determination, sculling effects in velocity determination, and scrolling effects in position determination). The bulk of the computations are contained in the basic operations that can be structured based on closed-form exact integral solutions to the Section 2 differential equations. Use of exact closed-form solutions for the basic operations translates directly into computer integration algorithm forms that are easily verified by simple and direct simulation techniques (Ref. 26).

3.1 Attitude

The classical exact integral solution to the Section 2.1 direction cosine attitude rate equation is as follows (Ref. 20 (or 25) Sects. 7.1.1, 7.1.1.1, and 7.1.1.2):

$$C_{B_m}^{N_{m-1}} = C_{B_{m-1}}^{N_{m-1}} \, C_{B_{I(m)}}^{B_{I(m-1)}}$$

$$C_{B_m}^{N_m} = C_{N_{I(m-1)}}^{N_{I(m)}} \, C_{B_m}^{N_{m-1}}$$

$$C_{B_{I(m)}}^{B_{I(m-1)}} = I + f_1(\phi_m)\left(\underline{\phi}_m\times\right) + f_2(\phi_m)\left(\underline{\phi}_m\times\right)^2 \qquad (8)$$

$$C_{N_{I(m-1)}}^{N_{I(m)}} = I - f_1(\zeta_m)\left(\underline{\zeta}_m\times\right) + f_2(\zeta_m)\left(\underline{\zeta}_m\times\right)^2$$

$$f_1(\chi) \equiv \frac{\sin\chi}{\chi} \qquad f_2(\chi) \equiv \frac{1-\cos\chi}{\chi^2}$$

where

m = System computer cycle time index for basic navigation parameter updating.

B_m, N_m = Coordinate Frame B and N orientations at navigation computer cycle time m.

$B_{I(m)}$, $N_{I(m)}$ = Discrete orientation of the B and N Frames in non-rotating inertial space (I) at computer cycle time t_m.

I = Identity matrix.

$\underline{\phi}_m$, $\underline{\zeta}_m$ = Rotation vector equivalents to the $C_{B_{I(m)}}^{B_{I(m-1)}}$ and $C_{N_{I(m-1)}}^{N_{I(m)}}$ direction cosine matrices (See Reference 20 (or 25) Section 3.2.2 for rotation vector definition).

ϕ_m, ζ_m = Magnitudes of $\underline{\phi}_m$, $\underline{\zeta}_m$.

χ = Dummy angle parameter.

Reference 20 (or 25) Sections 7.1.2, 7.1.2.1 and 7.1.2.2 provide the equivalent quaternion formulation integral solution which also is a function of the identical $\underline{\phi}_m$, $\underline{\zeta}_m$ rotation vectors.

Under constant inertial angular rates of the B and N Frames ($\underline{\omega}_{IB}^{B}$ and $\underline{\omega}_{IN}^{N}$), the $\underline{\phi}_m$, $\underline{\zeta}_m$ rotation vectors equal the simple integral of the B and N Frame inertial angular rates over the t_{m-1} to t_m time interval. Under dynamic angular rate conditions, $\underline{\phi}_m$, $\underline{\zeta}_m$ contain small additional "coning" terms that account for dynamic variations. The computation of $\underline{\phi}_m$ and $\underline{\zeta}_m$ is discussed in Section 3.4.

All of Equations (8) are analytically exact under general dynamic angular-rate conditions. An important point to recognize is that both direction cosine and quaternion based attitude algorithms have exact solutions using the identical ϕ_m, ζ_m rotation vector inputs. Hence, contrary to outdated popular belief, modern day quaternion and direction cosine attitude algorithm formulations have equal accuracy.

3.2 Velocity

The velocity algorithm implemented in the navigation software can be formulated from the integral of Equation (6) using a trapezoidal integration approximation for the small and/or slowly varying terms (Ref. 20 (or 25) Sects. 7.2, 7.2.2, 7.2.2.2 and 7.2.2.2.1 - note correction to Equation (7.2.2-4)):

$$\underline{v}_m^N = \underline{v}_{m-1}^N + \Delta\underline{v}_{SF_m}^N + \Delta\underline{v}_{G/COR_m}^N$$

$$\Delta\underline{v}_{G/COR_m}^N = \int_{t_{m-1}}^{t_m} \dot{\underline{v}}_{G/COR}^N \, dt \approx \frac{1}{2}\left(3\,\dot{\underline{v}}_{G/COR_{m-1}}^N - \dot{\underline{v}}_{G/COR_{m-2}}^N\right) T_m$$

$$\dot{\underline{v}}_{G/COR}^N \equiv \underline{g}^N - \underline{\omega}_{IE}^N \times \left(\underline{\omega}_{IE}^N \times \underline{R}^N\right) - \left(\underline{\omega}_{IN}^N + \underline{\omega}_{IE}^N\right) \times \underline{v}^N$$

$$\Delta\underline{v}_{SF_m}^N = \frac{1}{2}\left(C_{N_{I(m-1)}}^{N_{I(m)}} + I\right)\Delta\underline{v}_{SF_m}^{N_{m-1}} \approx \frac{1}{2}\left(2\,C_{N_{I(m-2)}}^{N_{I(m-1)}} - C_{N_{I(m-3)}}^{N_{I(m-2)}} + I\right)\Delta\underline{v}_{SF_m}^{N_{m-1}} \tag{9}$$

$$\Delta\underline{v}_{SF_m}^{N_{m-1}} = C_{B_{m-1}}^{N_{m-1}}\,\Delta\underline{v}_{SF_m}^{B_{m-1}}$$

$$\Delta\underline{v}_{SF_m}^{B_{m-1}} = \int_{t_{m-1}}^{t_m} C_{B_{I(t)}}^{B_{I(m-1)}}\,\underline{a}_{SF}^B\,dt = \left[I + f_2(\phi_m)\left(\underline{\phi}_m\times\right) + f_3(\phi_m)\left(\underline{\phi}_m\times\right)^2\right]\underline{\eta}_m$$

$$C_{B_{I(t)}}^{B_{I(m-1)}} = I + \int_{t_{m-1}}^{t} C_{B_{I(t)}}^{B_{I(m-1)}}\left(\underline{\omega}_{IB}^B\times\right)d\tau \qquad f_3(\chi) \equiv \frac{1}{\chi^2}\left(1 - \frac{\sin\chi}{\chi}\right)$$

where

$B_{I(t)}$ = B Frame orientation in non-rotating inertial space at time t after t_{m-1}.

$\Delta\underline{v}_{SF_m}$ = Velocity change from computer cycle m-1 to m due to specific force acceleration.

$\Delta\underline{v}_{G/COR_m}$ = Velocity change from computer cycle m-1 to m due to gravity and Coriolis acceleration. The approximate form shown is an extrapolation based on past (not yet updated) values of velocity and position.

$\underline{\eta}_m$ = Velocity translation vector from computer cycle m-1 to m.

t = General time in navigation.

τ = Dummy time parameter.

The approximate form shown for $\Delta \underline{v}_{SF_m}^{N}$ is based on $C_{N_{I(m-1)}}^{N_{I(m)}}$ (part of the Equations (8) with (18) attitude computations) being updated following the velocity and position update.

The $\Delta \underline{v}_{SF_m}^{B_{m-1}}$ expression in Equations (9) utilizes a velocity translation vector $\underline{\eta}_m$ (analogous to the rotation vector $\underline{\phi}_m$) to generate an analytically exact solution for $\Delta \underline{v}_{SF_m}^{B_{m-1}}$ under general dynamic angular-rate/specific-force conditions. The velocity translation vector concept was introduced by the author in Reference 23 as part of a unified framework for strapdown attitude/velocity/position integration algorithm formulation. Under constant B Frame specific force and inertial angular rate ($\underline{a}_{SF}^{B}$ and $\underline{\omega}_{IB}^{B}$), the $\underline{\eta}_m$ velocity translation vector equals the simple integral of B Frame specific force over the t_{m-1} to t_m time interval. Under dynamic angular-rate/specific-force conditions, $\underline{\eta}_m$ contains a small additional "sculling" term that accounts for dynamic variations. The computation of $\underline{\eta}_m$ is discussed in Section 3.4.

Except for trapezoidal integration error in the small and/or slowly varying terms, all of Equations (9) are analytically exact under general dynamic angular-rate/specific-force conditions.

3.3 Position

The position algorithm implemented in the navigation software can be formulated from the integral of Equations (7) using an extrapolated trapezoidal integration approximation for the small and/or slowly varying terms (Ref. 20 (or 25) Sects. 7.3.1, 7.3.3 and 7.3.3.1 - note correction to Equations (7.3.3-4)):

$$h_m = h_{m-1} + \Delta h_m$$

$$C_{N_{E(m)}}^{E} = C_{N_{E(m-1)}}^{E} \, C_{N_{E(m)}}^{N_{E(m-1)}}$$

$$C_{N_{E(m)}}^{N_{E(m-1)}} = I + f_1\left(\underline{\xi}_m\right) + f_2\left(\underline{\xi}_m\right)\left(\underline{\xi}_m \times\right)\left(\underline{\xi}_m \times\right)$$

$$\underline{\xi}_m \approx \int_{t_{m-1}}^{t_m} \underline{\rho}^{N}\, dt \approx \frac{1}{2}\left[\left(3\,\rho_{ZN_{m-1}} - \rho_{ZN_{m-2}}\right)\underline{u}_{Up}^{N}\, T_m + \left(3\, F_{C_{m-1}}^{N} - F_{C_{m-2}}^{N}\right)\left(\underline{u}_{Up}^{N} \times \Delta \underline{R}_m^{N}\right)\right]$$

$$\Delta h_m = \underline{u}_{Up}^{N} \cdot \Delta \underline{R}_m^{N}$$

$$\Delta \underline{R}_m^{N} \equiv \int_{t_{m-1}}^{t_m} \underline{v}^{N}\, dt \approx \left(\underline{v}_{m-1}^{N} + \frac{1}{2}\Delta \underline{v}_{G/COR_m}^{N}\right) T_m + \Delta \underline{R}_{SF_m}^{N} \tag{10}$$

$$\Delta \underline{R}_{SF_m}^{N} = \frac{1}{6}\left(C_{N_{m-1}}^{N_m} - I\right)\Delta \underline{v}_{SF_m}^{N_{m-1}}\, T_m + C_{B_{m-1}}^{N_{m-1}}\, \Delta \underline{R}_{SF_m}^{B_{m-1}}$$

$$\approx \frac{1}{6}\left(2\, C_{N_{m-2}}^{N_{m-1}} - C_{N_{m-3}}^{N_{m-2}} - I\right)\Delta \underline{v}_{SF_m}^{N_{m-1}}\, T_m + C_{B_{m-1}}^{N_{m-1}}\, \Delta \underline{R}_{SF_m}^{B_{m-1}}$$

$$\Delta \underline{R}_{SF_m}^{B_{m-1}} = \int_{t_{m-1}}^{t} \int_{t_{m-1}}^{\tau} C_{B_I(\tau_1)}^{B_{I(m-1)}}\, \underline{a}_{SF}^{B}\, d\tau_1\, d\tau = \left[I + 2\, f_3(\phi_m)\left(\underline{\phi}_m \times\right) + 2\, f_4(\phi_m)\left(\underline{\phi}_m \times\right)^2\right]\underline{\kappa}_m$$

$$f_4(\chi) \equiv \frac{1}{\chi^2}\left(\frac{1}{2} - \frac{(1 - \cos\chi)}{\chi^2}\right)$$

where

$N_{E(m)}$ = Discrete orientation of the N Frame in rotating earth space (E) at computer cycle time t_m.

$\underline{\xi}_m$ = Rotation vector equivalent to the $C_{N_{E(m)}}^{N_{E(m-1)}}$ direction cosine matrix. The computation is an extrapolated trapezoidal approximation to the exact integral of $\dot{\underline{\xi}}$ over an m cycle (similar to the Section 3.4 Equation (18) approximation for the integral of $\dot{\underline{\zeta}}$ in Equation (11), but using $\underline{\rho}^N$ in place of $\underline{\omega}_{IN}^N$).

ξ_m = Magnitude of $\underline{\xi}_m$.

$\underline{\zeta}_m$ = Calculated in Section 3.4 Equations (18).

Δh_m = Altitude change from computer cycle m-1 to m.

$\Delta \underline{R}_m$ = Position vector change from computer cycle m-1 to m.

$\Delta \underline{R}_{SF_m}$ = Specific force acceleration contribution to $\Delta \underline{R}_m$.

$\underline{\kappa}_m$ = Position translation vector from cycle m-1 to m.

The $\Delta \underline{R}_{SF_m}^{B_{m-1}}$ expression in Equations (10) utilizes a position translation vector $\underline{\kappa}_m$ (analogous to the rotation vector $\underline{\phi}_m$) to generate an analytically exact solution for $\Delta \underline{R}_{SF_m}^{B_{m-1}}$ under general dynamic angular-rate/specific-force conditions. The position translation vector concept was introduced by the author in Reference 23 as part of a unified framework for strapdown attitude/velocity/position integration algorithm formulation. Under constant B Frame specific force and inertial angular rate ($\underline{a}_{SF}^B$ and $\underline{\omega}_{IB}^B$), the $\underline{\kappa}_m$ position translation vector equals the simple double integral of B Frame specific force over the t_{m-1} to t_m time interval. Under dynamic angular-rate/specific-force conditions, $\underline{\kappa}_m$ contains a small additional "scrolling" term that accounts for dynamic variations. The computation of $\underline{\kappa}_m$ is discussed in Section 3.4.

Except for trapezoidal integration error in the small and/or slowly varying terms, all of Equations (10) are analytically exact under general dynamic angular-rate/specific-force conditions.

3.4 Computing The Rotation And Translation Vectors

The form of the $C_{B_{I(m)}}^{B_{I(m-1)}}$, $C_{N_{I(m-1)}}^{N_{I(m)}}$ expressions in (8) can be derived as the exact solution to Equations (1) under constant B and N Frame inertial angular rate (Ref. 20 (or 25) Sects. 3.2.2 and 3.2.2.1). The result would be identical to (8), but with the rotation vectors replaced by the integrals of the B and N Frame inertial rotation rates. Similarly, the forms of the $\Delta \underline{v}_{SF_m}^{B_{m-1}}$ and $\Delta \underline{R}_{SF_m}^{B_{m-1}}$ expressions in (9) and (10) can be derived as the exact analytic solution to the integrals in these expressions under constant B Frame inertial angular rate and specific force (Refs. 19 and 20 (or 25) Sects. 7.2.2.2 and 7.3.3). The result would be identical to the $\Delta \underline{v}_{SF_m}^{B_{m-1}}$ and $\Delta \underline{R}_{SF_m}^{B_{m-1}}$ expressions in (9) and (10), but with the rotation vector replaced by integrated B Frame angular rate and the velocity/position translation vectors replaced by the integral and double integral of B Frame specific force. In fact, the $\Delta \underline{v}_{SF_m}^{B_{m-1}}$ and $\Delta \underline{R}_{SF_m}^{B_{m-1}}$ expressions in (9) and (10) were derived in Reference 23 as the aforementioned exact solution under constant B Frame angular-rate/specific-force solution, but for general motion having the integrated B Frame angular rate term replaced by the rotation vector and the integrated/doubly-integrated B Frame specific force terms replaced by the translation vectors. This is the same approach used by Jordan in Reference 8 for introducing the $C_{B_{I(m)}}^{B_{I(m-1)}}$ expression in

(8) (which has been extended in this paper to also include $C_{N_{I(m-1)}}^{N_{I(m)}}$). For the Jordan case, the rotation vector was formulated by approximation as integrated angular rate plus a coning correction based on the Goodman-Robinson theorem (Ref. 4). The rotation vector concept was introduced by Euler and utilized by Laning in 1949 (Ref. 10) to develop the classical exact rotation vector rate of change equation (shown subsequently in this section) for strapdown inertial navigation application. Note: In 1971 Bortz reintroduced and applied the exact Laning rotation vector rate equation in a strapdown system/software implementation (Ref. 1) for which it has since been known as the "Bortz equation".

The integral of the Laning rotation vector rate equation provides an exact solution for the rotation vector input to the $C_{B_{I(m)}}^{B_{I(m-1)}}$, $C_{N_{I(m-1)}}^{N_{I(m)}}$ expressions in (8). Based on the previous discussion, the velocity/ position translation vectors $\underline{\eta}_m$, $\underline{\kappa}_m$ can be analytically defined as the vectors that satisfy the $\Delta\underline{v}_{SF_m}^{B_{m-1}}$ expression in (9) and the $\Delta\underline{R}_{SF_m}^{B_{m-1}}$ expression in (10). Using this definition, References 23 or 25 (Section 19.1.5) derive analytically exact equations for the translation vector rates of change (shown subsequently) which, when integrated from time t_{m-1} to t_m, provide exact solutions for $\underline{\eta}_m$ and $\underline{\kappa}_m$. References 23, 25 Sect. 19.1, and 20 (or 25) Sect. 7.1.1.2 then show that the following simplified forms can be utilized as accurate approximations for the $\dot{\underline{\phi}}$, $\dot{\underline{\zeta}}$, $\dot{\underline{\eta}}$ and $\dot{\underline{\kappa}}$ rotation/translation vector rates (Ref. 23 Equations (31) or Ref. 25 Equations (19.1.8-3), and Ref. 20 (or 25) Equation (7.1.1.2-4)):

$$\dot{\underline{\phi}} \approx \underline{\omega}_{IB}^{B} + \frac{1}{2}\,\underline{\alpha}(t) \times \underline{\omega}_{IB}^{B} \qquad \underline{\alpha}(t) \equiv \int_{t_{m-1}}^{t} \underline{\omega}_{IB}^{B}\, d\tau$$

$$\dot{\underline{\zeta}} \approx \underline{\omega}_{IN}^{N}$$

$$\dot{\underline{\eta}} \approx \underline{a}_{SF}^{B} + \frac{1}{2}\left(\underline{\alpha}(t) \times \underline{a}_{SF}^{B} - \underline{\omega}_{IB}^{B} \times \underline{\upsilon}(t)\right) \qquad \underline{\upsilon}(t) \equiv \int_{t_{m-1}}^{t} \underline{a}_{SF}^{B}\, d\tau \tag{11}$$

$$\dot{\underline{\kappa}} = \underline{\eta}(t) + \frac{1}{6}\left(\underline{\alpha}(t) \times \underline{\upsilon}(t) - 2\, \underline{\omega}_{IB}^{B} \times \underline{S}_{\upsilon}(t)\right) \qquad \underline{S}_{\upsilon}(t) \equiv \int_{t_{m-1}}^{t} \underline{\upsilon}\, d\tau$$

The error in the Equations (11) approximation is minimized by using a small value for the computer update cycle time interval t_{m-1} to t_m, thereby assuring small values of $\underline{\phi}$ and $\underline{\zeta}$. Using Equations (1) for $\underline{\omega}_{IN}^{N}$ with a trapezoidal integration algorithm (Ref. 20 (or 25) Sect. 7.1.1.2.1), the integral of Equations (11) over a computer update cycle then becomes for the rotation/translation vector inputs to Equations (8), (9) and (10):

$$\underline{\phi}_m = \underline{\alpha}_m + \Delta\underline{\phi}_{Cone_m} \qquad \underline{\eta}_m = \underline{\upsilon}_m + \Delta\underline{\eta}_{Scul_m} \qquad \underline{\kappa}_m = \underline{S}_{\upsilon_m} + \Delta\underline{\kappa}_{Scrl_m} \tag{12}$$

$$\Delta\underline{\phi}_{Cone_m} = \frac{1}{2}\int_{t_{m-1}}^{t_m} \left(\underline{\alpha}(t) \times \underline{\omega}_{IB}^{B}\right) dt \qquad \text{Coning} \tag{13}$$

$$\Delta\underline{\eta}_{Scul}(t) = \int_{t_{m-1}}^{t} \frac{1}{2}\left(\underline{\alpha}(\tau) \times \underline{a}_{SF}^{B} + \underline{\upsilon}(\tau) \times \underline{\omega}_{IB}^{B}\right) d\tau \qquad \Delta\underline{\eta}_{Scul_m} = \Delta\underline{\eta}_{Scul}(t_m) \qquad \text{Sculling} \tag{14}$$

$$\Delta\underline{\kappa}_{Scrl_m} = \frac{1}{6}\int_{t_{m-1}}^{t_m}\left(6\,\Delta\underline{\eta}_{Scul}(t) + \underline{\alpha}(t)\times\underline{\upsilon}(t) - 2\,\underline{\omega}_{IB}^{B}\times\underline{S}_{\upsilon}(t)\right)dt \qquad \text{Scrolling} \qquad (15)$$

$$\underline{S}_{\upsilon}(t) = \int_{t_{m-1}}^{t}\underline{\upsilon}(\tau)\,d\tau \qquad \underline{S}_{\upsilon_m} = \underline{S}_{\upsilon}(t_m) \qquad \begin{matrix}\text{Doubly integrated}\\ \text{specifice force acceleration}\end{matrix} \qquad (16)$$

$$\underline{\alpha}(t) = \int_{t_{m-1}}^{t}\underline{\omega}_{IB}^{B}\,d\tau \qquad \underline{\upsilon}(t) = \int_{t_{m-1}}^{t}\underline{a}_{SF}^{B}\,d\tau \qquad \begin{matrix}\underline{\alpha}_m = \underline{\alpha}(t_m)\\ \underline{\upsilon}_m = \underline{\upsilon}(t_m)\end{matrix} \qquad \begin{matrix}\text{Integrated inertial}\\ \text{sensor inputs}\end{matrix} \qquad (17)$$

$$\underline{\zeta}_m \approx \int_{t_{m-1}}^{t_m}\underline{\omega}_{IN}^{N}\,dt \approx \frac{1}{2}\left[\underline{\omega}_{IE_{m-1}}^{N} + \underline{\omega}_{IE_m}^{N} + \left(\rho_{ZN_{m-1}} + \rho_{ZN_m}\right)\underline{u}_{Up}^{N}\right]T_m + \frac{1}{2}\left(F_{C_{m-1}}^{N} + F_{C_m}^{N}\right)\left(\underline{u}_{Up}^{N}\times\Delta\underline{R}_m^{N}\right) \qquad \Delta\underline{R}_m^{N} \equiv \int_{t_{m-1}}^{t_m}\underline{v}^{N}\,dt \qquad (18)$$

where

- T_m = Time interval between m cycle updates.
- t_m = Time t at computer cycle m.
- $\underline{\alpha}_m$ = Integrated sensed B Frame angular rate vector from computer cycle m-1 to m.
- $\Delta\underline{\phi}_{Cone_m}$ = Coning contribution to $\underline{\phi}_m$.
- $\underline{\upsilon}_m$ = Integrated sensed B Frame specific force vector from computer cycle m-1 to m.
- $\Delta\underline{v}_{Scul_m}$ = Sculling contribution to $\underline{\eta}_m$.
- $\underline{S}_{\upsilon_m}$ = Doubly integrated sensed B Frame specific force vector from computer cycle m-1 to m.
- $\Delta\underline{\kappa}_{Scrl_m}$ = Scrolling contribution to $\underline{\kappa}_m$.

The $\Delta\underline{R}_m^{N}$ term in (18) is calculated as part of position updating operations (See Section 3.3, Equation (10)). The approximate form shown for $\underline{\zeta}_m$ is based on position being updated before attitude.

The $\Delta\underline{\phi}_{Cone_m}$ term in (13) has been coined the "coning" term because it measures the effect of "coning motion" components present in $\underline{\omega}_{IB}^{B}$. "Coning motion" is defined as the condition when an angular rate vector is itself rotating. For $\underline{\omega}_{IB}^{B}$ exhibiting pure coning motion (the $\underline{\omega}_{IB}^{B}$ magnitude being constant but the vector rotating) a fixed axis in the B Frame that is approximately perpendicular to the plane of the rotating $\underline{\omega}_{IB}^{B}$ vector will generate a conical surface in the I Frame as the angular rate motion ensues (hence, the term "coning" to describe the motion). Under coning angular motion conditions, B Frame axes perpendicular to $\underline{\omega}_{IB}^{B}$ appear to oscillate (in contrast with non-coning or "spinning" angular motion in which axes perpendicular to $\underline{\omega}_{IB}^{B}$ rotate around $\underline{\omega}_{IB}^{B}$). Note that the neglected terms in the $\dot{\underline{\zeta}}$ equation can also be identified as coning associated with the $\underline{\omega}_{IN}^{N}$ rate vector.

The $\Delta\underline{\eta}_{Scul_m}$ term in Equations (14), denoted as "sculling", measures the "constant" contribution to $\underline{\eta}_m$ created by combined dynamic angular-rate/specific-force rectification. The rectification is a maximum under classical sculling motion defined as sinusoidal angular-rate/specific-force in which the $\underline{\alpha}(t)$ angular excursion about one B Frame axis is at the same frequency and in phase with the $\underline{a}_{SF}^{B}$ specific force along another B Frame axis (with a constant acceleration component then produced along the average third axis direction). This is the same principle used by mariners to propel a boat in the forward direction using a single oar operated with an undulating motion (also denoted as "sculling", the original use of the term).

The $\Delta\underline{\kappa}_{Scrl_m}$ term in (15), denoted as "scrolling", is analogous to sculling in the velocity translation vector update equations. It measures the "constant" contribution to $\underline{\kappa}_m$ created by combined dynamic angular-rate/specific-force rectification. (The term "scrolling" was coined by the author merely to have a name for the term and also to have one that sounds like "sculling", but for position integration - change in the position vector $\underline{R}$ stressing the "R" sound. The complex mathematical formulations that accompany "scrolling" may be a more appropriate reason for the name). For all but the most exacting positioning applications, $\Delta\underline{R}_{Scrl_m}$ can be safely neglected.

Equations (11) (the basis for Equations (12) - (18)) are approximate forms of the following exact rotation/translation vector rate equations (Ref. 10, Ref. 20 (and 25) Sect. 7.1.1.1, Ref. 23, Equations (15) - (16) and Ref. 25 Sect. 19.1.5):

$$\dot{\underline{\phi}} = \underline{\omega}_{IB}^{B} + \frac{1}{2}\,\underline{\phi}\times\underline{\omega}_{IB}^{B} + f_5(\phi)\,\underline{\phi}\times\left(\underline{\phi}\times\underline{\omega}_{IB}^{B}\right) \qquad \dot{\underline{\zeta}} = \underline{\omega}_{IN}^{N} + \frac{1}{2}\,\underline{\zeta}\times\underline{\omega}_{IN}^{N} + f_5(\zeta)\,\underline{\zeta}\times\left(\underline{\zeta}\times\underline{\omega}_{IN}^{N}\right)$$

$$\begin{aligned}
\dot{\underline{\eta}} = {} & \underline{a}_{SF}^{B} + \frac{1}{2}\left(\underline{\phi}\times\underline{a}_{SF} - \dot{\underline{\phi}}\times\underline{\eta}\right) + f_5(\phi)\,\underline{\phi}\times\left(\underline{\phi}\times\underline{a}_{SF}^{B} - \dot{\underline{\phi}}\times\underline{\eta}\right) + f_3(\phi)\left(\underline{\phi}\times\dot{\underline{\phi}}\right)\times\underline{\eta} \\
& + \frac{1}{2}\,f_3(\phi)\left\{\left[\underline{\phi}\times\left(\underline{\phi}\times\dot{\underline{\phi}}\right)\right]\times\underline{\eta} - \underline{\phi}\cdot\underline{\eta}\;\underline{\phi}\times\dot{\underline{\phi}}\right\} + f_6(\phi)\left[\underline{\phi}\cdot\left(\dot{\underline{\phi}}\times\underline{\eta}\right)\right]\underline{\phi} - f_7(\phi)\,\underline{\phi}\cdot\underline{\eta}\;\underline{\phi}\times\left(\underline{\phi}\times\dot{\underline{\phi}}\right)
\end{aligned} \qquad (19)$$

$$\begin{aligned}
\dot{\underline{\kappa}} = {} & \underline{\eta} + \frac{1}{6}\left(\underline{\phi}\times\underline{\eta} - 2\,\dot{\underline{\phi}}\times\underline{\kappa}\right) + f_8(\phi)\,\underline{\phi}\times\left(\underline{\phi}\times\underline{\eta} - 2\,\dot{\underline{\phi}}\times\underline{\kappa}\right) + 2\,f_4(\phi)\left(\underline{\phi}\times\dot{\underline{\phi}}\right)\times\underline{\kappa} \\
& - f_9(\phi)\,\underline{\phi}\times\left[\left(\underline{\phi}\times\dot{\underline{\phi}}\right)\times\underline{\kappa}\right] - f_{10}(\phi)\,\phi^2\left(\underline{\phi}\times\underline{\eta} - 2\,\dot{\underline{\phi}}\times\underline{\kappa}\right) + f_{11}(\phi)\,\underline{\phi}\cdot\left(\dot{\underline{\phi}}\times\underline{\kappa}\right)\underline{\phi} \\
& + f_{12}(\phi)\left[\underline{\phi}\times\left(\underline{\phi}\times\dot{\underline{\phi}}\right)\right]\times\underline{\kappa} - f_{13}(\phi)\,\underline{\phi}\cdot\underline{\kappa}\;\underline{\phi}\times\left(\underline{\phi}\times\dot{\underline{\phi}}\right)
\end{aligned}$$

with

$$f_5(\chi) \equiv \frac{1}{\chi^2}\left(1 - \frac{\chi\sin\chi}{2\left(1-\cos\chi\right)}\right) \qquad f_6(\chi) \equiv \frac{1}{\chi^2}\left(1 - \frac{1}{2}\,f_1(\chi) - f_2(\chi)\right) \qquad f_7(\chi) \equiv \frac{1}{\chi^4}\left(f_1(\chi) + 2\,f_5(\chi)\,\chi^2 - 1\right)$$

$$f_8(\chi) \equiv f_3(\chi) - f_2(\chi)\,h_1(\chi) + f_1(\chi)\,h_2(\chi) \qquad f_9(\chi) \equiv 2\,f_4(\chi)\,h_1(\chi)$$

$$f_{10}(\chi) \equiv \frac{1}{(\chi)^2}\left[f_1(\chi)\,h_1(\chi) - f_2(\chi)\left(1 - h_2(\chi)\,(\chi)^2\right) + \frac{1}{6}\right] \qquad f_{11}(\chi) \equiv 2\left(2\,f_4(\chi)\,h_1(\chi) - f_3(\chi)\,h_2(\chi)\right)$$

$$f_{12}(\chi) \equiv \frac{1}{(\chi)^4}\,\frac{2\,h_1(\chi)}{f_3(\chi)}\left[1 - f_1^2(\chi) - f_2(\chi)\left(f_2(\chi) + f_3(\chi)\right)\phi^2\right] \qquad (20)$$

$$f_{13}(\chi) \equiv \frac{1}{(\chi)^2}\,\frac{2\,h_1(\chi)}{f_3(\chi)}\left[2\,f_3^2(\chi)\left(f_2(\chi) + 1\right) - f_2(\chi)\,f_4(\chi)\left(3 + 2\,f_2(\chi)\right)\right]$$

$$h_1(\chi) \equiv \frac{f_3(\chi)}{2\left(f_2^2(\chi) + f_3^2(\chi)\,(\chi)^2\right)} \qquad h_2(\chi) \equiv \frac{f_3^2(\chi) - f_2(\chi)\,f_4(\chi)}{f_2^2(\chi) + f_3^2(\chi)\,\phi^2(\chi)}$$

and

$$\underline{\phi}(t) = \int_{t_{m-1}}^{t} \dot{\underline{\phi}}(\tau)\, d\tau \qquad \underline{\phi}_m = \underline{\phi}(t_m) \qquad\qquad \underline{\zeta}(t) = \int_{t_{m-1}}^{t} \dot{\underline{\zeta}}(\tau)\, d\tau \qquad \underline{\zeta}_m = \underline{\zeta}(t_m)$$

$$\underline{\eta}(t) = \int_{t_{m-1}}^{t} \dot{\underline{\eta}}(\tau)\, d\tau \qquad \underline{\eta}_m = \underline{\eta}(t_m) \qquad\qquad \underline{\kappa}(t) = \int_{t_{m-1}}^{t} \dot{\underline{\kappa}}(\tau)\, d\tau \qquad \underline{\kappa}_m = \underline{\kappa}(t_m) \tag{21}$$

It is to be noted that the (19) with (20) translation vector rate equations are exact simplified analytically equivalent versions of Reference 23, Equations (15) - (16) (based on refined analysis since publication of Reference 23) - However, Equations (19) and (20) are identical to Reference 25, Equations (19.1.5-7) which were updated after publication of Reference 23. Note also that the $\dot{\underline{\eta}}$, $\dot{\underline{\kappa}}$ translation vector rates in (19) are functions of $\underline{a}_{SF}^{B}$ and rotation vector rate $\dot{\underline{\phi}}$ which is a function of inertial angular rate $\underline{\omega}_{IB}^{B}$. In Reference 27 using dual-quaternion/screw-vector theory, Wu shows that the velocity translation vector rate is analytically equivalent to the following further simplified exact version which is a function of $\underline{a}_{SF}^{B}$ and angular rate $\underline{\omega}_{IB}^{B}$ rather than $\dot{\underline{\phi}}$:

$$\dot{\underline{\eta}} = \underline{a}_{SF}^{B} + \frac{1}{2}\left(\underline{\phi} \times \underline{a}_{SF} - \underline{\omega}_{IB}^{B} \times \underline{\eta}\right) + f_5(\phi)\left[\underline{\phi} \times \left(\underline{\phi} \times \underline{a}_{SF}^{B} - \underline{\omega}_{IB}^{B} \times \underline{\eta}\right) - \left(\underline{\phi} \times \dot{\underline{\omega}}_{IB}^{B}\right) \times \underline{\eta}\right] + f_{14}(\phi)\, \underline{\phi} \cdot \underline{\eta}\, \underline{\phi} \times \left(\underline{\phi} \times \underline{\omega}_{IB}^{B}\right)$$

with (22)

$$f_{14}(\chi) \equiv \frac{\chi + \sin\chi}{2\,\chi^3 \left(1 - \cos\chi\right)} - \frac{2}{\chi^4}$$

As of this writing, a further simplified version of the exact position translation vector rate equation in (19) has yet to be found (Ref. 24).

Equations (19) - (22) are analytically exact under general angular-rate/specific-force dynamic conditions. It is easily verified by inspection that under constant B and N Frame inertial angular rate and constant B Frame specific force, the rotation/translation vectors reduce identically to the integrals of the first term in their respective rate equations (i.e., integrated $\underline{\omega}_{IB}^{B}$, $\underline{\omega}_{IN}^{N}$ for $\underline{\phi}$, $\underline{\zeta}$, integrated $\underline{a}_{SF}^{B}$ for $\underline{\eta}$, and doubly integrated $\underline{a}_{SF}^{B}$ for $\underline{\kappa}$), as they should in light of the discussion at the beginning of this section on their derivation. The additional terms in these equations (i.e., coning, sculling and scrolling) are small contributions excited by dynamic high frequency inputs (e.g., vibration), and not by lower frequency dynamic inputs that impact the leading terms. For example, in a 7.6 g root-mean-square aircraft vibration environment, Reference 20 (or 25), Section 7.4 shows that coning/sculling rates on the sensor assembly could be 9.9 deg/hr and 1.3 milli-gs worst case for a typically mounted INS (compared to lower frequency dynamic maneuver angular-rates/accelerations (e.g., 200 deg/sec and 10 gs) impacting the leading terms). Because the coning/sculling/scrolling terms are small, they can be accurately approximated by simplified versions of these terms in Equations (19) - (20). The principal benefit afforded by the use of rotation/translation vectors in structuring general strapdown navigation equations is that their rate equations can thereby be drastically simplified with virtually negligible error (Ref. 23). The utility of the exact rotation/translation rate representations in (19) - (22) is to provide a valid exact base from which to formulate simplified versions (e.g., Equations (11)) used for subsequent algorithm development, and as a reference for accuracy assessment of the simplified versions (Ref. 23).

3.5 Summary of Main Terms Requiring Integration Algorithms

Equations (8), (9) and (10) with (12) - (18) are integral solutions to Equations (3), (6) and (7) over a computer update cycle. For the most part, they consist of exact closed form expressions fed by the integrated sensor output terms in Equations (13) - (17). The $\underline{\alpha}$, $\underline{\upsilon}$ integrated angular rate and specific force acceleration signals in (17) (measured by summing (integrating) angular rate sensor and accelerometer integrated output increments) are the normal basic inputs to most strapdown inertial system algorithms. The Equations (13) - (16) terms (coning, sculling, scrolling, doubly integrated accelerometer signals) represent functions to be implemented by high speed digital computation algorithms operating within the basic m cycle update period.

4. DIGITAL INTEGRATION ALGORITHMS

Digital algorithms in the strapdown system computer are structured to provide integral solutions to the Section 2 differential equations based on repetitive processing at a specified computation rate. The integral solutions in Section 3 to the Section 2 equations have such a repetitive processing structure, hence, for the most part, are the digital algorithm forms to be programmed directly in the strapdown computer. These are exact solution forms, hence, have no algorithm error if programmed as shown (except for minor trapezoidal integration algorithm errors for the small/slowly varying terms). Exceptions are the coning, sculling, scrolling and doubly integrated sensor signal integrals in Section 3.4, Equations (13 - (16) needing high speed digital integration algorithms for implementation. The high speed algorithm errors are a function of the high speed digital integration update frequency. Additionally, Taylor series expansion algorithms are needed for the trigonometric function coefficients in Equations (8), (9) and (10) that avoid singularities when ϕ_m or ζ_m are near zero. Taylor series truncation error can be designed to be negligible by carrying sufficient terms.

Integration algorithms for the coning, sculling, scrolling and doubly integrated sensor signal terms are typically designed based on assumed approximate forms for the angular rate and specific force acceleration history during the computer update period. Commonly assumed forms for $\underline{\omega}_{IB}^{B}$ and $\underline{a}_{SF}^{B}$ are general polynomials in time:

$$\begin{aligned} \underline{\omega}_{IB}^{B} &= \underline{A}0_l + \underline{A}1_l\left(t - t_{l-1}\right) + \underline{A}2_l\left(t - t_{l-1}\right)^2 + \cdots \\ \underline{a}_{SF}^{B} &= \underline{B}0_l + \underline{B}1_l\left(t - t_{l-1}\right) + \underline{B}2_l\left(t - t_{l-1}\right)^2 + \cdots \end{aligned} \qquad (23)$$

where

l = High speed computer cycle time index for high speed digital integration algorithms (within the slower m cycles).

$\underline{A}i_l$, $\underline{B}i_l$ = Coefficient vectors selected to match the $\underline{\omega}_{IB}^{B}$ and $\underline{a}_{SF}^{B}$ signals from computer cycle l-1 to l.

The high speed updating algorithms can be structured based on truncated versions of Equations (23). The advantage of this approach is that the resulting digital algorithms are easily validated by simulation testing using the truncated forms they have been designed for as inputs. The algorithm solution should match the equivalent result obtained by analytical evaluation of the Section 3.4, Equation (11) integrals under the same truncated polynomial inputs (Ref. 26 and Ref. 20 (or 25) Sect. 11.1). Exact numerical correspondence should be the result for correctly structured and programmed algorithms.

Subsections to follow describe coning, sculling, scrolling and doubly integrated sensor signal digital integration algorithms designed to exactly match the Section 3.4, Equations (11) continuous integrals under Equations (23) polynomial inputs truncated after the $\underline{A}_1$ and $\underline{B}_1$ terms. Based on the discussion in the previous paragraph, Reference 26 Section 2.3 describes specialized simulators for validating algorithms of this structure. Following subsections also discuss singularity free algorithms for computing the $f_1(\chi)$ - $f_4(\chi)$ trigonometric functions in Sections 3.1-3.3 and whether orthogonality/normalization corrections are needed for the attitude algorithms.

4.1 Coning Digital Integration Algorithm

A coning digital computation algorithm for Equation (13) is given by (Ref. 20 (or 25) Sect. 7.1.1.1.1):

$$\Delta\underline{\phi}_{Cone_m} = \sum_l \left[\frac{1}{2}\left(\underline{\alpha}_{l-1} + \frac{1}{6}\Delta\underline{\alpha}_{l-1}\right) \times \Delta\underline{\alpha}_l\right] \quad \text{From } t_{m-1} \text{ to } t_m$$

$$\underline{\alpha}_l = \sum_l \Delta\underline{\alpha}_l \quad \text{From } t_{m-1} \text{ to } t_l \qquad \Delta\underline{\alpha}_l = \int_{t_{l-1}}^{t_l} d\underline{\alpha} \tag{24}$$

where

$\Delta\underline{\alpha}_l$ = Summation of integrated angular rate sensor output increments from cycle *l*-1 to *l*.

Equations (24) have been designed to be exact under Equations (23) angular rate input with the $\underline{\omega}_{IB}^{B}$ polynomial truncated after the $\underline{A}_1$ term.

4.2 Sculling Digital Integration Algorithm

A sculling digital computation algorithm for Equation (14) is given by (Ref. 20 (or 25) Sect. 7.2.2.2.2):

$$\Delta\underline{\eta}_{Scul_m} = \Delta\underline{\eta}_{Scul_l} \quad \text{At } t_m$$

$$\Delta\underline{\eta}_{Scul_l} = \sum_l \frac{1}{2}\left[\left(\underline{\alpha}_{l-1} + \frac{1}{6}\Delta\underline{\alpha}_{l-1}\right) \times \Delta\underline{\upsilon}_l + \left(\underline{\upsilon}_{l-1} + \frac{1}{6}\Delta\underline{\upsilon}_{l-1}\right) \times \Delta\underline{\alpha}_l\right] \quad \text{From } t_{m-1} \text{ to } t_l \tag{25}$$

$$\underline{\upsilon}_l = \sum_l \Delta\underline{\upsilon}_l \quad \text{From } t_{m-1} \text{ to } t_l \qquad \Delta\underline{\upsilon}_l = \int_{t_{l-1}}^{t_l} d\underline{\upsilon}$$

where

$\Delta\underline{\upsilon}_l$ = Summation of integrated accelerometer output increments from cycle *l*-1 to *l*.

Equations (25) have been designed to be exact under Equations (23) angular rate and specific force inputs with the $\underline{\omega}_{IB}^{B}$, $\underline{a}_{SF}^{B}$ polynomials truncated after the $\underline{A}_1$, $\underline{B}_1$ terms.

Note the similarity in form between the Equations (24) coning algorithm and Equations (25) sculling algorithm. Reference 14 provides a general formula for deriving the equivalent sculling algorithm (e.g., Equations (25)) from a previously derived coning algorithm (e.g., Equations (24)).

4.3 Scrolling And Doubly Integrated Sensor Signal Algorithms

Digital algorithms for scrolling computation and doubly integrated sensor signals for Equations (15) - (16) are given by Reference 25, Equations (19.1.11-1) (based on a similar development in Ref. 20 (or 25) Sect. 7.3.3.2 for an alternative scrolling formula):

$$\Delta\underline{\kappa}_{Scrl_m} = \sum_l \left(\delta\underline{\kappa}_{ScrlA_l} + \delta\underline{\kappa}_{ScrlB_l}\right) \qquad \text{From } t_{m-1} \text{ to } t_m$$

$$\delta\underline{\kappa}_{ScrlA_l} = \Delta\underline{\eta}_{Scul_{l-1}}\, T_l + \frac{1}{2}\left[\underline{\alpha}_{l-1} - \frac{1}{12}\left(\Delta\underline{\alpha}_l - \Delta\underline{\alpha}_{l-1}\right)\right] \times \left(\Delta\underline{S}_{\upsilon_l} - \underline{\upsilon}_{l-1}\, T_l\right)$$
$$+ \frac{1}{2}\left[\underline{\upsilon}_{l-1} - \frac{1}{12}\left(\Delta\underline{\upsilon}_l - \Delta\underline{\upsilon}_{l-1}\right)\right] \times \left(\Delta\underline{S}_{\alpha_l} - \underline{\alpha}_{l-1}\, T_l\right)$$

$$\delta\underline{\kappa}_{ScrlB_l} = \frac{1}{3}\left(\underline{S}_{\upsilon_{l-1}} - \frac{1}{8}\,\Delta\underline{\upsilon}_l\, T_l\right) \times \Delta\underline{\alpha}_l + \frac{1}{6}\left(\underline{\alpha}_{l-1} - \frac{3}{4}\,\Delta\underline{\alpha}_l + \frac{1}{4}\,\Delta\underline{\alpha}_{l-1}\right) \times \left(\underline{\upsilon}_{l-1} + \frac{5}{12}\,\Delta\underline{\upsilon}_l + \frac{1}{12}\,\Delta\underline{\upsilon}_{l-1}\right) T_l$$
$$+ \frac{1}{6}\left(\underline{\alpha}_{l-1} - \frac{3}{4}\,\Delta\underline{\alpha}_l + \frac{1}{4}\,\Delta\underline{\alpha}_{l-1}\right) \times \left(\underline{\upsilon}_{l-1} + \frac{5}{12}\,\Delta\underline{\upsilon}_l + \frac{1}{12}\,\Delta\underline{\upsilon}_{l-1}\right) T_l \qquad (26)$$

$$\Delta\underline{S}_{\alpha_l} = \underline{\alpha}_{l-1} T_l + \frac{T_l}{12}\left(5\,\Delta\underline{\alpha}_l + \Delta\underline{\alpha}_{l-1}\right) \qquad \Delta\underline{S}_{\upsilon_l} = \underline{\upsilon}_{l-1} T_l + \frac{T_l}{12}\left(5\,\Delta\underline{\upsilon}_l + \Delta\underline{\upsilon}_{l-1}\right)$$

$$\underline{S}_{\upsilon_l} = \sum_l \Delta\underline{S}_{\upsilon_l} \qquad \text{From } t_{m-1} \text{ to } t_l \qquad \underline{S}_{\upsilon_m} = \underline{S}_{\upsilon_l} \quad \text{at } t_m$$

where

T_l = Time interval between computer high speed l cycles.

Equations (26) have been designed to be exact under Equations (23) angular rate and specific force inputs with the $\underline{\omega}_{IB}^{B}$, $\underline{a}_{SF}^{B}$ polynomials truncated after the $\underline{A}_1$, $\underline{B}_1$ terms.

4.4 Trigonometric Coefficient Algorithms

To assure that no singularities occur when ϕ_m or ζ_m are near zero, the following Taylor series expansion formulas can be used for the Equations (8), (9) and (10) $C_{BI(m)}^{BI(m-1)}$, $C_{NI(m-1)}^{NI(m)}$, $\Delta\underline{v}_{SF_m}^{B_{m-1}}$, $\Delta\underline{R}_{SF_m}^{B_{m-1}}$, trigonometric function coefficients:

$$f_1(\chi) = \frac{\sin\chi}{\chi} = 1 - \frac{\chi^2}{3\,!} + \frac{\chi^4}{5\,!} - \cdots \qquad f_2(\chi) = \frac{(1 - \cos\chi)}{\chi^2} = \frac{1}{2\,!} - \frac{\chi^2}{4\,!} + \frac{\chi^4}{6\,!} - \cdots \qquad (27)$$
$$f_3(\chi) = \frac{1}{\chi^2}\left(1 - \frac{\sin\chi}{\chi}\right) = \frac{1}{3\,!} - \frac{\chi^2}{5\,!} + \frac{\chi^4}{7\,!} - \cdots \qquad f_4(\chi) = \frac{1}{\chi^2}\left(\frac{1}{2} - \frac{(1 - \cos\chi)}{\chi^2}\right) = \frac{1}{4\,!} - \frac{\chi^2}{6\,!} + \frac{\chi^4}{8\,!} - \cdots$$

Corresponding computational algorithms are then structured from truncated versions of the former. The series can be truncated with a sufficient number of terms to assure "error free" performance. For example, to assure overall eleventh order accuracy in $C_{BI(m)}^{BI(m-1)}$ (Equations (8)), this would entail carrying $f_1(\chi)$ out to tenth order (in ϕ_m) and $f_2(\chi)$ out to eighth order (note, there is no ninth order term in $f_2(\chi)$).

4.5 Orthogonality and Normalization Algorithms

Orthogonality and normalization correction algorithms can be applied to computed direction cosine matrices (e.g., C_B^N and C_N^E) to preserve the proper characteristics of their rows and columns (Ref. 20 (or 25) Sect. 7.1.1.3). Similarly, normalization algorithms can be applied to quaternion attitude representations

(Ref. 20 (or 25) Sect. 7.1.2.3). One of the advantages in using exact formulated attitude updating algorithms (e.g., Equations (8)) is that direction cosines and equivalent quaternion formulations calculated by integration, will remain orthogonal and normal if initialized as such, independent of sensor error (Ref. 20 (or 25) Sect. 3.5.1). Consequently, if computer register round-off error is negligible (as it is for most applications using modern day processors), there is no need for orthogonality/normality compensation.

5. STRAPDOWN SENSOR ERROR COMPENSATION

A fundamental problem with all inertial navigation systems is the inability to manufacture inertial components with the inherent accuracy required to meet system requirements. To correct for this deficiency, compensation algorithms are included in the INS software for correcting sensor outputs for known predictable error effects. The compensation algorithms represent the inverse of the inertial sensor analytical model equations.

This section describes error models and compensation algorithms that can be used to correct for errors in the strapdown inertial sensors (angular rate sensors and accelerometers), relative displacement between accelerometers ("size effect"), misalignment of the strapdown sensor assembly relative to the system mount, and alignment of the system mount in the user vehicle relative to vehicle reference axes. Included is a discussion of the application of the sensor compensation algorithms to the Section 4 strapdown inertial navigation integration routines and their associated coning, sculling, scrolling and accelerometer size-effect/anisoinertia elements.

5.1 Sensor Error Models

This section characterizes the errors typically present in the raw inertial sensor outputs (angular rate sensors and accelerometers) and then describes a general form of compensation equations for correcting the errors. All vectors in this section are represented in the B Frame, the designation for which has been omitted for analytical simplicity.

The output vector from strapdown angular rate sensor and accelerometer triads can be characterized as a function of their inputs as (Ref. 20 (or 25) Sects. 8.1.1.1 and 8.1.1.2):

$$\begin{aligned} \underline{\omega}_{IB_{Puls}} &= \frac{1}{\Omega_{Wt_0}} \left(I + F_{Scal} \right) \left(F_{Algn}\, \underline{\omega}_{IB} + \delta\underline{\omega}_{Bias} + \delta\,\underline{\omega}_{Quant} + \delta\underline{\omega}_{Rand} \right) \\ \underline{a}_{SF_{Puls}} &= \frac{1}{A_{Wt_0}} \left(I + G_{Scal} \right) \left(G_{Algn}\, \underline{a}_{SF} + \delta\underline{a}_{Bias} + \delta\,\underline{a}_{Size} + \delta\,\underline{a}_{Aniso} + \delta\underline{a}_{Quant} + \delta\underline{a}_{Rand} \right) \end{aligned} \qquad (28)$$

where

$\underline{\omega}_{IB_{Puls}}$, $\underline{a}_{SF_{Puls}}$ = Angular rate sensor and accelerometer triad output vector in pulses per second. Each axis output pulse is a digital indication that the sensor associated with that axis has received an integrated input increment equal to that particular sensor's pulse size.

Ω_{Wt_0}, A_{Wt_0} = Nominal pulse weight (a positive value) for each angular rate sensor (radians per pulse) and accelerometer (fps per pulse).

F_{Scal}, G_{Scal} = Angular rate sensor and accelerometer triad scale factor correction matrices; diagonal matrices in which each element adjusts the output pulse scaling to correspond to the actual scaling for the particular sensor output. May include non-linear scale factor effects and temperature dependency. Nominally, F_{Scal} and G_{Scal} are zero.

F_{Algn}, G_{Algn} = Alignment matrices for the angular rate sensor and accelerometer triads. Each row represents a unit vector along a particular sensor input axis as projected onto the B-Frame. May include specific force acceleration dependency. Nominally, F_{Algn} and G_{Algn} are identity.

$\delta\underline{\omega}_{Bias}$, $\delta\underline{a}_{Bias}$ = Angular rate sensor and accelerometer triad bias vectors. Each element equals the systematic output from a sensor under zero input conditions. May have environmental sensitivities (e.g., temperature, specific force acceleration for angular rate sensors, angular rate for accelerometers).

$\delta\underline{\omega}_{Quant}$, $\delta\underline{a}_{Quant}$ = Instantaneous angular rate sensor and accelerometer triad pulse quantization error vectors associated with the output only being provided when the cumulative input equals the pulse weight per axis.

$\delta\underline{\omega}_{Rand}$, $\delta\underline{a}_{Rand}$ = Angular rate sensor and accelerometer triad random error output vectors.

$\delta\underline{a}_{Size}$ = Accelerometer triad size effect error created by the fact that due to physical size, the accelerometers in the triad cannot be collocated, hence, do not measure components of identically the same acceleration vector.

$\delta\underline{a}_{Aniso}$ = Accelerometer triad anisoinertia error effect (present in pendulous accelerometers) created by mismatch in the moments of inertia around the input and pendulum axes.

References 21 and 20 (or 25) Section 8.1.3 analytically describe the Equations (28) $\delta\underline{\omega}_{Quant}$, $\delta\underline{a}_{Quant}$ quantization error effects in strapdown inertial sensors. The $\delta\underline{a}_{Size}$ size effect term (Ref. 20 (or 25) Sect. 8.1.4.1) and for pendulous accelerometers, the $\delta\underline{a}_{Aniso}$ anisoinertia term (Ref. 16 and Ref. 20 (or 25) Sect. 8.1.4.2), are given by :

$$\delta\underline{a}_{Size} \equiv \sum_{k=1,3} \left\{ \underline{G}^T_{Algn_k} \cdot \left[\dot{\underline{\omega}}_{IB} \times \underline{l}_k + \underline{\omega}_{IB} \times \left(\underline{\omega}_{IB} \times \underline{l}_k \right) \right] \right\} \underline{u}_k$$

$$\delta\underline{a}_{Aniso} = K_{Aniso} \sum_{k=1,3} \omega_{IB_k}\, \omega_{IB_{kp}}\, \underline{u}_k \tag{29}$$

where

$\underline{u}_k$ = Unit vector parallel to the accelerometer k input axis.

$\underline{l}_k$ = Position vector from INS navigation center to accelerometer k center of seismic mass.

$\underline{G}^T_{Algn_k}$ = Vector formed from the k^{th} column of G^T_{Algn}, the transpose of the G_{Algn} accelerometer triad alignment matrix

K_{Aniso} = Accelerometer anisoinertia coefficient (a generic property of the accelerometer design).

ω_{IB_k}, $\omega_{IB_{kp}}$ = Angular rate $\underline{\omega}_{IB}$ projections on the accelerometer k input and kp pendulum axes.

5.2 Generic Strapdown Sensor Compensation Forms

The inverse of Equations (28) form the basis for compensating the $\underline{\omega}_{IB_{Puls}}$, $\underline{a}_{SF_{Puls}}$ raw sensor outputs to calculate the true $\underline{\omega}_{IB}$, $\underline{a}_{SF}$ angular-rate/specific-force-acceleration inputs for the strapdown inertial integration operations (Ref. 20 (or 25) Sects. 8.1.1.1 and 8.1.1.2). First, Equations (28) are solved for the B Frame angular rate and acceleration input vector:

$$\underline{\omega}'_{IB} = \Omega_{Wt_0} \left(I + F_{Scal} \right)^{-1} \underline{\omega}_{IB_{Puls}}$$

$$\underline{a}'_{SF} = A_{Wt_0} \left(I + G_{Scal} \right)^{-1} \underline{a}_{SF_{Puls}} \tag{30}$$

$$\underline{\omega}_{IB} = F_{Algn}^{-1} \left(\underline{\omega}'_{IB} - \delta\underline{\omega}_{Bias} - \delta\underline{\omega}_{Quant} - \delta\underline{\omega}_{Rand} \right)$$
$$\underline{a}_{SF} = G_{Algn}^{-1} \left(\underline{a}'_{SF} - \delta\underline{a}_{Bias} - \delta\underline{a}_{Size} - \delta\underline{a}_{Aniso} - \delta\underline{a}_{Quant} - \delta\underline{a}_{Rand} \right) \tag{31}$$

where

$\underline{\omega}'_{IB}, \underline{a}'_{SF}$ = Scale factor compensated angular rate sensor and accelerometer output vectors.

Equations (30) represent the scale factor compensation equation for the raw angular rate sensor and accelerometer triad $\underline{\omega}_{IBPuls}$, $\underline{a}_{SFPuls}$ outputs. Compensation for the remaining predictable errors in $\underline{\omega}_{IBPuls}$ and $\underline{a}_{SFPuls}$ is achieved using a simplified form of (31) in which it is recognized that the $\delta\underline{\omega}_{Rand}$ and $\delta\underline{a}_{Rand}$ components are unpredictable, hence, can only be approximated by zero:

$$\underline{\omega}_{IB} \approx F_{Algn}^{-1} \left(\underline{\omega}'_{IB} - \delta\underline{\omega}_{Bias} - \delta\underline{\omega}_{Quant} \right)$$
$$\underline{a}_{SF} = G_{Algn}^{-1} \left(\underline{a}'_{SF} - \delta\underline{a}_{Bias} - \delta\underline{a}_{Size} - \delta\underline{a}_{Aniso} - \delta\underline{a}_{Quant} \right) \tag{32}$$

Compensation Equations (32) are further refined to a more familiar form by introducing the following definitions:

$$\Omega_{Wt} \equiv \Omega_{Wt_0} \left(I + F_{Scal} \right)^{-1} \qquad A_{Wt} \equiv A_{Wt_0} \left(I + G_{Scal} \right)^{-1}$$
$$K_{Mis} \equiv I - F_{Algn}^{-1} \qquad L_{Mis} \equiv I - G_{Algn}^{-1} \tag{33}$$
$$\underline{K}_{Bias} \equiv F_{Algn}^{-1} \, \delta\underline{\omega}_{Bias} \qquad \underline{L}_{Bias} \equiv G_{Algn}^{-1} \, \delta\underline{a}_{Bias}$$

Substituting (33) into (30) and (32) obtains the equivalent compensation equations:

$$\underline{\omega}'_{IB} = \Omega_{Wt} \, \underline{\omega}_{IBPuls}$$
$$\underline{\omega}_{IB} \approx \underline{\omega}'_{IB} - K_{Mis} \, \underline{\omega}' - \underline{K}_{Bias} - F_{Algn}^{-1} \, \delta\underline{\omega}_{Quant}$$
$$\underline{a}'_{SF} = A_{Wt} \, \underline{a}_{SFPuls} \tag{34}$$
$$\underline{a}_{SF} \approx \underline{a}'_{SF} - L_{Mis} \, \underline{a}'_{SF} - \underline{L}_{Bias} - G_{Algn}^{-1} \left(\delta\underline{a}_{Size} + \delta\underline{a}_{Aniso} + \delta\underline{a}_{Quant} \right)$$

In many systems, the form of the compensation equations so derived contain linearization approximations to the exact inverse relations (to conserve on computer throughput). The approach taken above is the analytically simpler expedient of using the exact inverse of the complete error model (without linearization approximation) based on the assumption that modern day computers can easily handle the workload.

5.3 Generic Strapdown Sensor Compensation Algorithms

Equations (34) are the basis for the following algorithms used to form the inputs to the Section 3 navigation parameter m cycle updating operations (Ref. 20 (or 25) Sects. 8.1.2.1 and 8.1.2.2):

$$
\begin{aligned}
\underline{\alpha}'_m &= \Omega_{Wt}\,\underline{\alpha}_{Cnt_m} \\
\underline{\alpha}_m &\approx \underline{\alpha}'_m - K_{Mis}\,\underline{\alpha}'_m - \underline{K}_{Bias}\,T_m - \delta\underline{\alpha}_{QuantC_m} \\
\underline{S}'_{\alpha_m} &= \Omega_{Wt}\,\underline{S}_{\alpha Cnt_m} \\
\underline{S}_{\alpha_m} &\approx \underline{S}'_{\alpha_m} - K_{Mis}\,\underline{S}'_{\alpha_m} - \frac{1}{2}\left(\underline{K}_{Bias}\,T_m + \delta\underline{\alpha}_{QuantC_m}\right)T_m \\
\underline{\upsilon}'_m &= A_{Wt}\,\underline{\upsilon}_{Cnt_m} \\
\underline{\upsilon}_m &\approx \underline{\upsilon}'_m - L_{Mis}\,\underline{\upsilon}'_m - \underline{L}_{Bias}\,T_m - \delta\underline{\upsilon}_{SizeC_m} - \delta\underline{\upsilon}_{AnisoC_m} - \delta\underline{\upsilon}_{QuantC_m} \\
\underline{S}'_{\upsilon_m} &= A_{Wt}\,\underline{S}_{\upsilon Cnt_m} \\
\underline{S}_{\upsilon_m} &\approx \underline{S}'_{\upsilon_m} - L_{Mis}\,\underline{S}'_{\upsilon_m} - \frac{1}{2}\left(\underline{L}_{Bias}\,T_m + \delta\underline{\upsilon}_{SizeC_m} + \delta\underline{\upsilon}_{AnisoC_m} + \delta\underline{\upsilon}_{QuantC_m}\right)T_m
\end{aligned}
\tag{35}
$$

in which (with Equations (29)) the following definitions apply:

$$
\begin{aligned}
\delta\underline{\upsilon}_{SizeC_m} &\equiv G_{Algn}^{-1}\int_{t_{m-1}}^{t_m}\delta\underline{a}_{Size}\,dt \approx \int_{t_{m-1}}^{t_m}\delta\underline{a}_{Size}\,dt \\
&= \sum_k \int_{t_{m-1}}^{t_m}\left\{\underline{u}_k\cdot\left[\dot{\underline{\omega}}_{IB}\times\underline{l}_k + \underline{\omega}_{IB}\times\left(\underline{\omega}_{IB}\times\underline{l}_k\right)\right]\right\}\underline{u}_k\,dt \\
\delta\underline{\upsilon}_{AnisoC_m} &\equiv G_{Algn}^{-1}\int_{t_{m-1}}^{t_m}\delta\underline{a}_{Aniso}\,dt \approx \int_{t_{m-1}}^{t_m}\delta\underline{a}_{Aniso}\,dt = K_{Aniso}\sum_{k=1,3}\underline{u}_k\int_{t_{m-1}}^{t_m}\omega_{IB_k}\,\omega_{IB_{kp}}\,dt \\
\delta\underline{\alpha}_{QuantC_m} &\equiv F_{Algn}^{-1}\int_{t_{m-1}}^{t_m}\delta\underline{\omega}_{Quant}\,dt \approx \int_{t_{m-1}}^{t_m}\delta\underline{\omega}_{Quant}\,dt \\
\delta\underline{\upsilon}_{QuantC_m} &\equiv G_{Algn}^{-1}\int_{t_{m-1}}^{t_m}\delta\underline{a}_{Quant}\,dt \approx \int_{t_{m-1}}^{t_m}\delta\underline{a}_{Quant}\,dt
\end{aligned}
\tag{36}
$$

$$
\underline{\alpha}_{Cnt_m} \equiv \int_{t_{m-1}}^{t_m} d\underline{\alpha}_{Cnt} \qquad \underline{\upsilon}_{Cnt_m} \equiv \int_{t_{m-1}}^{t_m} d\underline{\upsilon}_{Cnt}
$$

Summation of raw sensor output pulses over computer cycle m

where

$d\underline{\alpha}_{Cnt}, d\underline{\upsilon}_{Cnt}$ = Angular rate sensor and accelerometer instantaneous pulse output vectors.

Reference 20 (or 25) Sect. 8.1.3 (and its subsections) describe various methods for calculating the $\delta\underline{\alpha}_{QuantC_m}$, $\delta\underline{\upsilon}_{QuantC_m}$ sensor quantization compensation terms. Representative algorithms for the $\delta\underline{\upsilon}_{SizeC_m}$, $\delta\underline{\upsilon}_{AnisoC_m}$ accelerometer size effect and anisoinertia compensation terms are described next.

5.3.1 Representative Accelerometer Size Effect And Anisoinertia Computation Algorithms

The size effect and anisoinertia terms in Equations (36) can be calculated at the high speed l cycle rate within each m cycle as follows (Ref. 20 (or 25) Sects. 8.1.4.1.1.1 and 8.1.4.2):

$$\beta_{ij_m} \equiv \sum_{l} \Delta\alpha_{i_l}\, \Delta\alpha_{j_l} \qquad \text{From } t_{m-1} \text{ to } t_m$$

$$\begin{aligned} \delta\upsilon_{SizeCY_m} = f_{Size} \Big[& - l_{Z_2}\left(\Delta\alpha_{X_m} - \Delta\alpha_{X_{m-1}}\right) + l_{X_2}\left(\Delta\alpha_{Z_m} - \Delta\alpha_{Z_{m-1}}\right) \\ & + l_{Z_2}\, \beta_{YZ_m} + l_{X_2}\, \beta_{XY_m} - l_{Y_2}\left(\beta_{ZZ_m} + \beta_{XX_m}\right)\Big] \end{aligned} \qquad (37)$$

$$\delta\upsilon_{SizeCZ_m}, \delta\upsilon_{SizeCX_m} = \text{Similarly by permuting subscripts.}$$

$$\delta\underline{\upsilon}_{AnisoC_m} = f_{Size}\, K_{Aniso} \sum_{k=1,3} \beta_{kp_m}\, \underline{u}_k$$

where

l_{ik} = Component of $\underline{l}_k$ along B Frame axis i.

f_{Size} = Size effect algorithm computation frequency which equals the reciprocal of T_l.

$\Delta\alpha_{i_l}$ = Integrated angular rate around B Frame axis i over the l-1 to l computer cycle time interval.

$\Delta\alpha_{i_m}, \Delta\alpha_{i_{m-1}}$ = $\Delta\alpha_{i_l}$ for the l-1 to l cycle time intervals immediately preceding the m and m-1 cycle times.

$\delta\upsilon_{SizeCi_m}$ = i^{th} B Frame component of $\delta\underline{\upsilon}_{SizeC_m}$.

The previous algorithm is designed to compute the high frequency dependent terms (β_{ij}) at the l cycle rate, use them to calculate size effect at the m cycle rate, and apply the size effect correction at the m cycle rate in Equations (35). This implies that size-effect compensation is not being applied at the l cycle rate, hence, will not be provided on the acceleration data used for high speed sculling calculations (Equations (25)). The associated sculling error is of the same order of magnitude as the basic Equations (37) size-effect correction, thus, cannot be ignored. Section 5.4 describes an algorithm for correcting the associated sculling error at the m cycle rate. Alternatively, the full Equations (37) size-effect correction can be computed and applied at the high speed l cycle rate with β_{ij_m} replaced by $\Delta\alpha_{i_l}\, \Delta\alpha_{j_j}$. The sculling computation would then be performed with the size-effect compensated accelerometer data, thereby eliminating the previously described sculling error.

5.4 Compensation Of High Speed Algorithms For Sensor Error

The high speed algorithms described in Sections 4.1- 4.3 and 5.3.1 for coning, sculling, scrolling, doubly integrated sensor signals, size effect and anisoinertia are based on error free values for the $\Delta\underline{\alpha}_l$ and $\Delta\underline{\upsilon}_l$ integrated angular rate sensor and accelerometer increment inputs. This implies that compensated sensor signals are being used, thereby implying sensor compensation to be performed at the l cycle rate in forming $\Delta\underline{\alpha}_l$ and $\Delta\underline{\upsilon}_l$. The equivalent result can also be obtained by performing the high speed computations with uncompensated sensor data, then compensating the result at the slower m cycle rate. A savings in throughput can thereby be achieved if needed for a particular application. For the coning algorithm, the associated operations would be as follows (Ref. 20 (or 25) Sect. 8.2.1.1):

$$\Delta\underline{\phi}_{ConeCnt_m} \equiv \int_{t_{m-1}}^{t_m} \frac{1}{2}\,\underline{\alpha}_{Cnt}(t) \times d\underline{\alpha}_{Cnt}$$

$$\Delta\underline{\phi}'_{Cone_m} = \Omega_{ConeWt}\,\Delta\underline{\phi}_{ConeCntm} \qquad \Delta\underline{\phi}_{Cone_m} = \left(I - K_{MisCone}\right)\Delta\underline{\phi}'_{Cone_m} \tag{38}$$

in which

$$K_{MisCone} \equiv \begin{bmatrix} \left(K_{Mis_{YY}} + K_{Mis_{ZZ}}\right) & -K_{Mis_{YX}} & -K_{Mis_{ZX}} \\ -K_{Mis_{XY}} & \left(K_{Mis_{ZZ}} + K_{Mis_{XX}}\right) & -K_{Mis_{ZY}} \\ -K_{Mis_{XZ}} & -K_{Mis_{YZ}} & \left(K_{Mis_{XX}} + K_{Mis_{YY}}\right) \end{bmatrix}$$

$$\Omega_{ConeWt} \equiv \begin{bmatrix} \Omega_{Wt_Y}\,\Omega_{Wt_Z} & 0 & 0 \\ 0 & \Omega_{Wt_Z}\,\Omega_{Wt_X} & 0 \\ 0 & 0 & \Omega_{Wt_X}\,\Omega_{Wt_Y} \end{bmatrix} \tag{39}$$

where

$\underline{\alpha}_{Cnt}(t) = \underline{\alpha}(t)$ as defined in Equations (11) but based on angular rate sensor output counts.

$\Omega_{Wt_i}, K_{Mis_{ij}}$ = Elements in row i of column i of Ω_{Wt} and row i column j of K_{Mis}.

Sensor compensation applied at the m cycle rate on uncompensated computed inputs to the accelerometer size effect and anisoinertia routines in Equations (37) would be (Ref. 20 (or 25) Sect. 8.1.4.1.4):

$$\beta_{ij_m} = \Omega_{Wt_i}\,\Omega_{Wt_j}\,\beta_{ijCnt_m} \qquad \Delta\alpha_{i_m} = \Omega_{Wt_i}\,\Delta\alpha_{iCnt_m} \tag{40}$$

where

$\beta_{ijCnt_m}, \Delta\alpha_{iCnt_m} = \beta_{ij_m}, \Delta\alpha_{i_m}$ computed with uncompensated sensor pulse output data.

Similar but more complicated operations are required for post *l* cycle sculling and scrolling compensation for sensor error (Ref. 20 (or 25) Sects. 8.2.2.1 and 8.2.3.1). In most applications, however, ignoring sensor misalignment effects in the sculling, scrolling (and size-effect/anisoinertia) calculations introduces negligible error. Based on this assumption, it then is reasonable to use the direct approach of performing scale factor compensation on the raw angular rate sensor and accelerometer input data (i.e., applying Ω_{Wt} and A_{Wt}) at the *l* cycle rate, and then applying the scale factor compensated signals as input to the sculling, scrolling (and accelerometer size effect/anisoinertia) *l* cycle computation algorithms (Equations (25), (25) and (37)). However, such an approach can still leave significant error in the sculling/scrolling computations executed using scale factor compensated sensor data without accelerometer size-effect compensation. Reference 20 (or 25), Section 8.1.4.1 shows that the residual sculling error can be accurately approximated and corrected with:

$$\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m} \approx \int_{t_{m-1}}^{t_m} \frac{1}{2}\left(\underline{\alpha}(t) \times \delta\underline{a}_{Size} + \delta\underline{\upsilon}_{SizeC}(t) \times \underline{\omega}_{IB}\right) dt \qquad \delta\underline{\upsilon}_{SizeC}(t) \approx \int_{t_{m-1}}^{t} \delta\underline{a}_{Size}\, d\tau \tag{41}$$

where

$\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}$ = Size effect correction to be applied to a $\Delta\underline{\eta}_{Scul_m}$ sculling term calculated with accelerometer data not containing size effect compensation.

The $\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}$ correction is applied at the m cycle rate by augmenting the translation vectors in Equations (12) as follows:

$$\underline{\eta}_m = \underline{\upsilon}_m + \Delta\underline{\eta}_{Scul_m} - \delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}$$
$$\underline{\kappa}_m = \underline{S}_{\upsilon_m} + \Delta\underline{\kappa}_{Scrl_m} - \frac{1}{2}\,\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}\,T_m \tag{42}$$

Reference 20 (or 25) Section 8.1.4.1.2 shows that $\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}$ in (41) can be accurately approximated by the following algorithm whose form and magnitude is similar to the basic Equation (37) size-effect compensation algorithm:

$$\begin{aligned}\delta\Delta\eta_{Scul\text{-}SizeCY_m} = f_{Size}\Big\{&\frac{1}{2}\alpha_{Z_m}\Big[\big(\Delta\alpha'_{Y_m} + \Delta\alpha'_{Y_{m-1}}\big)\,l_{Z_1} - \big(\Delta\alpha'_{Z_m} + \Delta\alpha'_{Z_{m-1}}\big)\,l_{Y_1}\Big]\\ &- \frac{1}{2}\alpha_{X_m}\Big[\big(\Delta\alpha'_{X_m} + \Delta\alpha'_{X_{m-1}}\big)\,l_{Y_3} - \big(\Delta\alpha'_{Y_m} + \Delta\alpha'_{Y_{m-1}}\big)\,l_{X_3}\Big]\\ &+ \beta_{XX_m}\,l_{Y_3} + \beta_{ZZ_m}\,l_{Y_1} - \beta_{XY_m}\,l_{X_3} - \beta_{YZ_m}\,l_{Z_1}\Big\}\end{aligned} \tag{43}$$

$\delta\Delta\eta_{Scul\text{-}SizeCZ_m}$, $\delta\Delta\eta_{Scul\text{-}SizeCX_m}$ = Similarly by permuting subscripts.

where

$\delta\Delta\eta_{Scul\text{-}SizeCi_m}$ = i^{th} B Frame component of $\delta\Delta\underline{\eta}_{Scul\text{-}SizeC_m}$.

$\Delta\alpha'_{i_m}$ = i^{th} component of $\Delta\alpha_{i_m}$ with only scale factor compensation.

α_{i_m} = i^{th} component of $\underline{\alpha}_m$.

The alternative to using (42) with (43) is to apply the Equations (37) size-effect compensation at the high speed l cycle rate to the scale factor compensated accelerometer data (i.e., using scale factor compensated $\Delta\underline{\alpha}_l$ angular rate sensor data for $\Delta\alpha_{i_m}$ with β_{ij_m} replaced by $\Delta\alpha_{i_l}\,\Delta\alpha_{j_j}$). The sculling computation would then be performed with the size-effect compensated accelerometer data, thereby eliminating the Equations (41) error effect.

5.5 Compensation For Sensor Triad Attitude Error

The K_{Mis} and L_{Mis} misalignment error compensation coefficients described in Section 5.2 represent misalignment of the strapdown sensor axes relative to nominally defined B Frame sensor coordinates. An additional misalignment to be compensated in the INS is misalignment of the nominal B Frame relative to the reference axes of the user vehicle in which the INS is installed.

The attitude of the vehicle in which the strapdown inertial navigation system (INS) is installed is determined from the attitude direction matrix C_B^N, inertial sensor assembly mounting misalignments (relative to the INS mount), and the orientation of the INS mount relative to user vehicle reference axes. An attitude direction cosine matrix relating the user vehicle and locally level attitude reference axes can be written as (Ref. 20 (or 25) Sect. 8.3):

$$C_{VRF}^N = C_B^N \left(C_B^M\right)^T C_{VRF}^M \tag{44}$$

where

M = INS mount coordinate frame (the B Frame is nominally aligned to the M Frame).

VRF = User vehicle reference axes.

The C_B^M direction cosine matrix can be defined without approximation in terms of the associated rotation vector components as follows:

$$C_B^M = I + \frac{\sin J}{J}\left(\underline{J}\times\right) + \frac{(1 - \cos J)}{J^2}\left(\underline{J}\times\right)^2 \qquad (45)$$

where

$\underline{J}$, J = Sensor triad mount misalignment rotation error vector and its magnitude.

The $\underline{J}$ components are compensation coefficients measured during system calibration (Ref. 20 (or 25) Sect. 18.4.7.4). The C_{VRF}^M matrix is a function of the particular mount orientation in the user vehicle.

6. STRAPDOWN INERTIAL NAVIGATION ERROR PROPAGATION EQUATIONS

The overall strapdown INS design process requires supporting analyses to develop and verify performance specifications. This generally entails the use of a strapdown INS error model in the form of time rate differential equations that describe the error response of INS computed attitude/velocity/position data. Such error models are also fundamental to the design of Kalman filters used, in conjunction with other system inputs, for correcting the INS errors. This section describes strapdown INS error model equations that represent the INS attitude/velocity/position navigation parameter integration routine response to sensor errors (i.e., excluding the effect of algorithm and computer finite word-length error, errors that are generally negligible in a well designed modern day INS compared to sensor error effects). The term "sensor error" used in this section refers to the residual error in the sensor signals after applying the Section 5 compensation corrections. It is only the residual sensor errors that generate INS navigation parameter output errors. The residual sensor errors arise from inaccuracy in measuring the sensor compensation coefficients, sensor random noise outputs that are not accounted for in the compensation algorithms, short and long term sensor instabilities, and variations in actual sensor performance from the analytical models in Section 5.1 that formed the basis for the sensor compensation algorithms.

6.1 Typical Strapdown Error Parameters

An important part of strapdown INS error model development is the definition (and selection) of attitude/velocity/position error parameters used in the error model and their relationship to the INS integration computed navigation parameters (or to a hypothetical set of INS navigation parameters that are analytically related to the INS computed set). The INS computed navigation parameters described in Sections 2 - 4 are the C_B^N matrix for attitude, the $\underline{v}^N$ vector for velocity, the C_N^E matrix for horizontal earth referenced position, and altitude h for vertical earth referenced position. These contain 20 individual scalar parameters, each of which develop errors in response to sensor error. Furthermore, the 18 error parameters associated with the C_B^N and C_N^E matrices (9 elements each) are not independent due to natural orthogonality/normality constraints that govern all direction cosine matrices. To circumvent the problem of dealing with the attendant complexities, navigation error is typically described in terms of three navigation error vectors (for attitude, velocity, and position), each consisting of three independent error components. The error in the INS computed navigation parameters (in this case, C_B^N, $\underline{v}^N$, C_N^E and h) are analytical functions of the independent error vector parameters. For example, the N Frame components of a commonly used set of attitude, velocity, and position error parameters is (Ref. 20 (or 25) Sects. 12.2.1-12.2.3 and 12.5) :

$$
\begin{aligned}
\left(\underline{\psi}^{N}\times\right) &\equiv C_E^N \left[I - \hat{C}_B^E \, C_E^B \right] C_N^E + \left[\left(C_B^N \, \delta\underline{\alpha}_{Quant}^B \right) \times \right] \\
\delta\underline{V}^N &\equiv C_E^N \left(\hat{\underline{v}}^E - \underline{v}^E \right) - C_B^N \, \delta\underline{\upsilon}_{Quant}^B \qquad (46) \\
\delta\underline{R}^N &\equiv C_E^N \left(\hat{\underline{R}}^E - \underline{R}^E \right) = R \left(C_E^N \, \hat{C}_N^E - I \right) \underline{u}_{Up}^N + \delta h \, \underline{u}_{Up}^N
\end{aligned}
$$

where

$\hat{}$ = Designator for a system computer calculated quantity containing error. The quantity without the $\hat{}$ designation is by definition error free (e.g., C_B^A is error free and $\hat{C}_B^A$ contains errors).

$\underline{\psi}$ = Small angle error rotation vector associated with the computed C_B^E attitude matrix.

$\delta\underline{V}$ = Error in the computed $\underline{v}$ velocity vector relative to the earth measured in the E Frame.

$\delta\underline{R}$ = Error in the computed position vector from earth's center $\underline{R}$ measured in the E Frame.

$\delta\underline{\alpha}_{Quant}$, $\delta\underline{\upsilon}_{Quant}$ = Angular rate sensor and accelerometer triad quantization error residual (remaining after applying quantization compensation - Ref. 20 (or 25) Sect. 8.1.3 and subsections).

The quantization terms in the $\underline{\psi}$ and $\delta\underline{V}$ equations are included to facilitate differential error equation modeling (See further explanation at conclusion of Section 6.2 to follow).

An equivalent set of attitude, velocity, position error parameters can also be defined that are more directly related to the C_B^N, $\underline{v}^N$, C_N^E, h navigation parameters computed by direct integration of Equations (1), (6) and (7) (previous references):

$$
\begin{aligned}
\left(\underline{\gamma}^{N}\times\right) &\equiv I - \hat{C}_B^N \, C_N^B + \left[\left(C_B^N \, \delta\underline{\alpha}_{Quant}^B \right) \times \right] \\
\delta\underline{v}^N &\equiv \hat{\underline{v}}^N - \underline{v}^N - C_B^N \, \delta\underline{\upsilon}_{Quant}^B \qquad (47) \\
\left(\underline{\varepsilon}^{N}\times\right) &\equiv C_E^N \, \hat{C}_N^E - I \\
\delta h &\equiv \hat{h} - h
\end{aligned}
$$

where

$\underline{\gamma}$ = Small angle error rotation vector in the computed C_B^N attitude matrix.

$\delta\underline{v}$ = Error in computed velocity measured in the N Frame.

$\underline{\varepsilon}$ = Small angle error rotation vector in the computed C_N^E position matrix.

δh = Error in computed altitude.

The two sets of navigation error parameters are analytically related through (previous references):

$$\underline{\psi}^N = \underline{\gamma}^N - \underline{\varepsilon}^N$$

$$\delta\underline{V}^N = \delta\underline{v}^N + \underline{\varepsilon}^N \times \underline{v}^N \qquad (48)$$

$$\delta\underline{R}^N = R\left(\underline{\varepsilon}^N \times \underline{u}^N_{Up}\right) + \delta h \, \underline{u}^N_{Up}$$

or the equivalent inverse relationships:

$$\underline{\varepsilon}^N = \frac{1}{R}\left(\underline{u}^N_{Up} \times \delta\underline{R}^N\right) + \varepsilon_{ZN} \, \underline{u}^N_{Up}$$

$$\delta h = \underline{u}^N_{Up} \cdot \delta\underline{R}^N$$

$$\delta\underline{v}^N = \delta\underline{V}^N - \underline{\varepsilon}^N \times \underline{v}^N \qquad (49)$$

$$\underline{\gamma}^N = \underline{\psi}^N + \underline{\varepsilon}^N$$

where

ε_{ZN} = Local vertical component of $\underline{\varepsilon}$ (projection on the N Frame Z axis along $\underline{u}_{Up}$).
R = Distance from earth's center to the current position location (magnitude of $\underline{R}$).

6.2 Inertial Sensor Error Parameters

Classical error models for the angular rate sensor and accelerometer triad outputs following compensation (in which the error in accelerometer size effect and anisoinertia compensation is ignored as negligible) are as follows (Ref. 20 (or 25) Sects. 12.4 - 12.5):

$$\delta\underline{\omega}^B_{IB} = \delta K_{Scal/Mis} \, \underline{\omega}^B_{IB} + \delta\underline{K}_{Bias} + \delta \, \underline{\omega}_{Rand}$$

$$\delta\underline{a}^B_{SF} = \delta L_{Scal/Mis} \, \underline{a}^B_{SF} + \delta\underline{L}_{Bias} + \delta \, \underline{a}_{Rand} \qquad (50)$$

where

$\delta\underline{\omega}^B_{IB}$, $\delta\underline{a}^B_{SF}$ = Angular rate sensor and accelerometer triad vector error residuals following sensor compensation but excluding $\delta\underline{\alpha}_{Quant}$, $\delta\underline{\upsilon}_{Quant}$ quantization compensation error residuals.

$\delta K_{Scal/Mis}$, $\delta L_{Scal/Mis}$ = Residual angular rate sensor and accelerometer scale-factor/misalignment error matrices remaining after applying Ω_{Wt}, K_{Mis}, A_{Wt}, L_{Mis} compensation in Equations (34).

$\delta\underline{K}_{Bias}$, $\delta\underline{L}_{Bias}$ = Residual angular rate sensor and accelerometer bias error vectors remaining after applying $\underline{K}_{Bias}$, $\underline{L}_{Bias}$ compensation in Equations (34).

Note that the $\delta\underline{\alpha}_{Quant}$, $\delta\underline{\upsilon}_{Quant}$ quantization compensation error residuals do not appear in the Equations (50) $\delta\underline{\omega}^B_{IB}$, $\delta\underline{a}^B_{SF}$ error definitions, but instead, show in the Equations (46) - (47) navigation parameter error vectors. Reference 21 and Reference 20 (or 25) Section 12.5 show that this form results in the navigation error parameter time rate propagation equations being in standard error state dynamic format (with quantization noise inputs appearing directly, not as their derivatives) as shown next.

6.3 Error Parameter Propagation Equations

The $\underline{\psi}$, $\delta\underline{V}$, $\delta\underline{R}$ error parameters defined in Section 6.1 propagate in N Frame coordinates as (Ref. 20 (or 25) Sects. 12.3.3 and 12.5.1):

$$\dot{\underline{\psi}}^{N} = -\,C_{B}^{N}\,\delta\underline{\omega}_{IB}^{B} - \underline{\omega}_{IN}^{N} \times \underline{\psi}^{N} + C_{B}^{N}\left(\underline{\omega}_{IB}^{B} \times \delta\underline{\alpha}_{Quant}\right)$$

$$\delta\dot{\underline{V}}^{N} = C_{B}^{N}\,\delta\underline{a}_{SF}^{B} + \underline{a}_{SF}^{N} \times \underline{\psi}^{N} - \frac{g}{R}\,\delta\underline{R}_{H}^{N} + F(h)\,\frac{g}{R}\,\delta R\,\underline{u}_{Up}^{N} - \left(\underline{\omega}_{IE}^{N} + \underline{\omega}_{IN}^{N}\right) \times \delta\underline{V}^{N} + \delta\underline{g}_{Mdl}^{N}$$
$$-\left(\underline{a}_{SF}^{N}\times\right) C_{B}^{N}\,\delta\underline{\alpha}_{Quant} - \left[\left(C_{B}^{N}\,\underline{\omega}_{IB}^{B} + \underline{\omega}_{IE}^{N}\right)\times\right] C_{B}^{N}\,\delta\underline{\upsilon}_{Quant}$$

$$F(h) = 2 \quad \text{For} \quad h \geq 0 \qquad\qquad F(h) = -1 \quad \text{For} \quad h < 0 \qquad\qquad (51)$$

$$\delta\dot{\underline{R}}^{N} = \delta\underline{V}^{N} - \underline{\omega}_{EN}^{N} \times \delta\underline{R}^{N} + C_{B}^{N}\,\delta\underline{\upsilon}_{Quant}$$

$$\delta\underline{R}_{H}^{N} = \delta\underline{R}^{N} - \delta R\,\underline{u}_{Up}^{N}$$

$$\delta R = \underline{u}_{Up}^{N} \cdot \delta\underline{R}^{N}$$

where

$\delta\underline{R}_{H}$, δR = Horizontal and upward vertical components of $\delta\underline{R}$.

$\delta\underline{g}_{Mdl}$ = Modeling error in $\underline{g}$ produced by variations in true gravity from the model used in the system computer.

Equations (51) are based on attitude/velocity/position being updated in the strapdown computer at the same algorithm repetition rate. For different repetition rates the quantization terms in these equations have revised coefficients. Note also that the vertical velocity error equations in (51) are different for positive compared to negative altitudes. This is a manifestation of the difference in gravity model below versus above the earth's surface (Ref. 20 (or 25) Sect. 5.4).

Equations (51) can be integrated to calculate the response of the attitude, velocity, position errors in a strapdown INS as impacted by accelerometer, angular rate sensor, and gravity model approximation errors. The equations are based on the assumption that the INS navigation parameter integration algorithm error and computer round-off error is negligibly small.

A similar set of N Frame error propagation equations exist for the Equations (47) $\underline{\gamma}$, $\delta\underline{v}$, $\underline{\varepsilon}$, δh error parameters (Ref. 20 (or 25) Sects. 12.3.4 and 12.5.2). Equations (51) for $\underline{\psi}$, $\delta\underline{V}$, $\delta\underline{R}$ and the equivalent set for $\underline{\gamma}$, $\delta\underline{v}$, $\underline{\varepsilon}$, δh can be derived from the differential of any set of strapdown inertial navigation error propagation equations (e.g., the set given in Section 2) with the appropriate definitions substituted for the navigation parameter error terms (e.g., Equations (46) or (47)). Alternatively, Reference 20 (or 25) Section 12.3.6 (and subsections) shows that one set of error parameter propagation equations can be derived from another by applying the equivalency equations relating the parameters (e.g., Equations (48) or (49)). It is important to recognize that the parameters selected to describe the error characteristics of a particular INS can be any convenient set and not necessarily those derived from the navigation parameter differential equations actually implemented in the INS software. Thus, any set of error propagation equations can be used to model the error characteristics of any INS, provided that the error propagation equations and INS navigation parameter integration algorithms are analytically correct without singularities over the range of interest, and that the sensor error models are appropriate for the application.

7. CONCLUDING REMARKS

Computational operations in strapdown inertial navigation systems are analytically traceable to basic time rate differential equations of rotational and translational motion as a function of angular-rate/specific-force-acceleration vectors and local gravitation. Modern day strapdown INS computer capabilities allow the use of navigation parameter integration algorithms based on exact solutions to the differential equations. This considerably simplifies the software validation process and can result in a single set of universal algorithms that can be used over a broad range of strapdown applications. Exact attitude updating algorithms based on direction cosines or an attitude quaternion are analytically equivalent with identical error characteristics that are a function of the error in the same computed attitude rotation vector inputs to each. Modern day strapdown computational algorithms and computer capabilities render the computational error negligible compared to sensor error effects.

The angular-rate/specific-force-acceleration vectors input to the strapdown INS digital integration algorithms are measured by angular rate sensors and accelerometers whose errors are compensated in the strapdown system computer based on classical error models for the inertial sensors. Strapdown INS attitude/velocity/position output errors are produced by errors remaining in the inertial sensor signals following compensation (due to sensor error model inaccuracies, sensor error instabilities, sensor calibration errors) and to gravity modeling errors. Resulting INS navigation error characteristics can be defined by various attitude, velocity, position error parameters that are analytically equivalent. Any set of navigation parameter error propagation equations can be used to predict the error performance of any strapdown INS. The navigation error parameters used in the error propagation equations do not have to be directly related to the navigation parameters used in the strapdown INS computer integration algorithms.

REFERENCES

1. Bortz J. E., “A New Mathematical Formulation for Strapdown Inertial Navigation”, *IEEE Transactions on Aerospace and Electronic Systems*, Volume AES-7, No. 1, January 1971, pp. 61-66.

2. Britting, K. R., *Inertial Navigation System Analysis*, John Wiley and Sons, New York, 1971.

3. “Department Of Defense World Geodetic System 1984”, NIMA TR8350.2, Third Edition, 4 July 1997.

4. Goodman, L.E. and Robinson, A.R., "Effects of Finite Rotations on Gyroscope Sensing Devices", *Journal of Applied Mechanics*, Vol. 25, June 1958.

5. Ignagni, M. B., “Optimal Strapdown Attitude Integration Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 13, No. 2, March-April 1990, pp. 363-369.

6. Ignagni, M. B., “Efficient Class Of Optimized Coning Compensation Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 19, No. 2, March-April 1996, pp. 424-429.

7. Ignagni, M. B., “Duality of Optimal Strapdown Sculling and Coning Compensation Algorithms”, *Journal of the ION*, Vol. 45, No. 2, Summer 1998.

8. Jordan, J. W., “An Accurate Strapdown Direction Cosine Algorithm”, NASA TN-D-5384, September 1969.

9. Kachickas, G. A., “Error Analysis For Cruise Systems”, *Inertial Guidance*, edited by Pitman, G. R., Jr., John Wiley & Sons, New York, London, 1962.

10. Laning, J.H., Jr., "The Vector Analysis of Finite Rotations and Angles", Massachusetts Institute of Technology, Cambridge, Instrumentation Laboratory Special Report 6398-S-3, 1949.

11. Litmanovich, Y. A., Lesyuchevsky, V. M. & Gusinsky, V. Z., “Two New Classes of Strapdown Navigation Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 23, No. 1, January- February 2000.

12. Mark, J.G. & Tazartes, D.A., “On Sculling Algorithms”, 3rd St. Petersburg International Conference On Integrated Navigation Systems, St. Petersburg, Russia, May 1996.

13. Miller, R., “A New Strapdown Attitude Algorithm”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 6, No. 4, July-August 1983, pp. 287-291.

14. Roscoe, K. M., “Equivalency Between Strapdown Inertial Navigation Coning and Sculling Integrals/Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 24, No. 2, March-April 2001, pp. 201-205.

15. Savage, P. G., “A New Second-Order Solution for Strapped-Down Attitude Computation”, AIAA/JACC Guidance & Control Conference, Seattle, Washington, August 15-17, 1966.

16. Savage, P. G., “Strapdown Sensors”, *Strapdown Inertial Systems - Theory And Applications*, NATO AGARD Lecture Series No. 95, June 1978, Section 2.

17. Savage, P. G., “Strapdown System Algorithms”, *Advances In Strapdown Inertial Systems*, NATO AGARD Lecture Series No. 133, May 1984, Section 3.

18. Savage, P. G., “Strapdown Inertial Navigation System Integration Algorithm Design Part 1 - Attitude Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 21, No. 1, January-February 1998, pp. 19-28.

19. Savage, P. G., “Strapdown Inertial Navigation System Integration Algorithm Design Part 2 - Velocity and Position Algorithms”, *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 21, No. 2, March-April 1998, pp. 208-221.

20. Savage, P. G., *Strapdown Analytics,* Strapdown Associates, Inc., Maple Plain, Minnesota, 2000.

21. Savage, P. G., "Analytical Modeling of Sensor Quantization in Strapdown Inertial Navigation Error Equations", *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 25, No. 5, September-October 2002, pp. 833-842.

22. Savage, P. G., "Strapdown System Computational Elements", *Advances In Navigation Sensors and Integration Technology*, NATO RTO Lecture Series No. 232, October 2003, Section 3.

23. Savage, P. G., "A Unified Mathematical Framework For Strapdown Algorithm Design", *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 29, No. 2, March-April 2006, pp. 237-249.

24. Savage, P. G., "Reply by the Author to Y. Wu", *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1485-1486.

25. Savage, P. G., *Strapdown Analytics - Second Edition,* Strapdown Associates, Inc., Maple Plain, Minnesota, 2007.

26. Savage, P. G., "Performance Analysis Of Strapdown Systems", *Low Cost Navigation Sensors and Integration Technology*, NATO RTO-EN-SET-116, October 2008, Section 4.

27. Wu, Y. "Comment on 'A Unified Mathermatical Framework for Strapdown Algorithm Design' ", *AIAA Journal Of Guidance, Control, And Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1482-1484.