# Performance Analysis Of Strapdown Systems

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

WBN-14011
www.strapdownassociates.com
June 2, 2015

Originlly published in
NATO Research and Technology Organization Organization (RTO)
Sensors and Electronics Technology Panel (SET)
Low-Cost Navigation Sensors and Integration Technology
RTO EDUCATIONAL NOTES RTO-SET-116(2008), Section 10
Published in 2009

## ABSTRACT

This paper provides an overview of assorted analysis techniques associated with strapdown inertial navigation systems. The process of strapdown system algorithm validation is discussed. Closed-form analytical simulator drivers are described that can be used to exercise/validate various strapdown algorithm groups. Analytical methods are presented for analyzing the accuracy of strapdown attitude, velocity and position integration algorithms (including position algorithm folding effects) as a function of algorithm repetition rate and system vibration inputs. Included is a description of a simplified analytical model that can be used to translate system vibrations into inertial sensor inputs as a function of sensor assembly mounting imbalances. Strapdown system static drift and rotation test procedures/equations are described for determining strapdown sensor calibration coefficients. The paper overviews Kalman filter design and covariance analysis techniques and describes a general procedure for validating aided strapdown system Kalman filter configurations. Finally, the paper discusses the general process of system integration testing to verify that system functional operations are performed properly and accurately by all hardware, software and interface elements.

## 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 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. The 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.

$\underline{\omega}_{A_1A_2}$ = Angular rate of Coordinate Frame $A_2$ relative to Coordinate Frame $A_1$. When $A_1$ is non-rotating, $\underline{\omega}_{A_1A_2}$ 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

An important part of strapdown inertial navigation system (INS) analysis deals with performance assessment of particular technology elements. One of the most common is covariance simulation analysis which determines the expected system errors based on statistical estimation. This paper discusses performance analysis methods which, although infrequently reported, are a fundamental part of the design and accuracy assessment of aided and unaided inertial systems: inertial computation algorithm validation, system vibration effects analysis, system testing for inertial sensor calibration error, and Kalman filter validation.

The primary computational elements in a strapdown inertial navigation system consist of integration operations for calculating attitude, velocity and position navigation parameters using strapdown angular rate and specific force acceleration for input. These operations are resident in the system computer and are comprised of computational algorithms designed to perform the required digital integration operations. An important part of the algorithm design is the validation process used to assure that the digital integration operations accurately create an attitude, velocity, position history corresponding to a continuous integration of time rate differential equations for the navigation parameters. Structuring the algorithms such that they are primarily based on exact closed-form solutions to the differential equations significantly simplifies the validation process, allowing it to be executed using simple closed-form exact solution reference truth models that are application independent. This paper provides examples of such truth models describing there use in validating representative strapdown algorithms.

The accuracy of well-structured strapdown computational algorithms is ultimately limited by their ability to perform their designated functions in the presence of sensor vibrations. The algorithm repetition rate is a determining factor in this regard which must be selected small enough to meet specified software accuracy requirements. This paper describes some simple analytical techniques for predicting strapdown inertial sensor dynamic motion and resulting algorithm error in the presence of angular/linear inertial sensor vibrations. Included is a description of a simplified sensor-assembly/mount structural dynamic analytical model for translating INS input vibration into strapdown sensor inputs.

Following inertial sensor calibration and strapdown inertial system final assembly, the system must be tested to verify proper performance and in the process, assess the residual calibration errors remaining in the inertial sensor compensation coefficients. The paper describes two commonly used system level tests, the Strapdown Drift Test (for measuring angular rate sensor bias residuals), and the Strapdown Rotation Test (for measuring angular-rate-sensor/accelerometer misalignment/scale-factor-error and accelerometer bias). Both tests are structured based on measurements from a stabilized "platform" created by software operations on the strapdown sensor signals. This method considerably reduces the accuracy requirements for rotation test fixtures used in the tests.

Kalman filtering has become the standard method for updating inertial system navigation parameters (and sensor compensation coefficients) during operation (i.e., the "aided" inertial navigation system configuration). A Kalman filter is a sophisticated set of software operations processed in parallel with the normal strapdown inertial navigation integration algorithms.

Proper operation of an aided inertial system depends on thorough validation of the Kalman filter software. Such a validation process is described in the paper based on a generic model of a real time Kalman filter. Included is an overview of covariance analysis techniques for assessing aided (and unaided) system performance on a statistical basis.

The paper concludes with a general discussion of system integration procedures to assure that all system hardware, software and associated interface elements function properly and accurately.

This paper is an updated version of Reference 7. Reference 7 is a condensed summary of material originally published in the two volume textbook *Strapdown Analytics* (Ref. 6), the second edition of which has been recently published (Reference 9). *Strapdown Analytics* provides a broad detailed exposition of the analytical aspects of strapdown inertial navigation technology. This version of the Reference 7 paper also incorporates new material from the recently published paper *A Unified Mathematical Framework For Strapdown Algorithm Design* (Reference 8) - also provided in Section 19.1 of the second edition of *Strapdown Analytics* (Reference 9). Equations in this paper (as in Reference 7) are presented without proof. Their derivations are provided in Reference 6 (or 9) as delineated throughout the paper by Reference 6 (or 9) section number (or by Reference 10 Equation number which, in Reference 10, are referenced to sections in Reference 6 (or 9) or equations in Reference 8 for their derivation source).

## 2. STRAPDOWN ALGORITHM VALIDATION

A key aspect of the strapdown inertial navigation software design process is validation of the digital integration algorithms. In general this consists of operating the integration algorithms in a test computer at their specified repetition rate with inertial sensor inputs provided by a "truth model" having a corresponding navigation parameter profile (e.g., attitude, velocity, position). The navigation parameter solution generated with the strapdown algorithms under test is compared numerically against the equivalent truth model profile parameters to validate the algorithms.

The success of the validation depends on the accuracy of the truth model navigation reference solution profile accompanying the truth model sensor data. Ideally, the reference solution should be completely error free with the attitude, velocity, position parameters representing an error free integration of the truth model inertial sensor signals. In addition, the reference solution profile(s) should be designed to exercise all elements of the computational algorithms under test. In general, this dictates reference profile(s) that do not represent realistic conditions encountered in normal navigation system use. It also generally involves several simulation profiles, each designed to exercise different groupings of the computational algorithms under test.

In general, two methods can be considered for the truth model; 1. A digital integration approach in which the truth model integration algorithms are more accurate than the INS integration algorithms being validated, and 2. Closed-form analytical equations representing exact integral solutions of the inertial sensor angular-rate/linear-acceleration inputs to the INS

integration algorithms. The problem with the Method 1 approach is the dilemma it presents in demonstrating the accuracy of a truth model that also contains digital integration algorithm error. This section addresses the Method 2 approach, and provides two examples from Reference 6 (or 9) of closed-form analytically exact truth models for evaluating classical groupings of INS algorithms used to execute basic integration operations; 1. Attitude updating under dynamic coning conditions, 2. Attitude updating, acceleration transformation, velocity/position updating under sculling/scrolling dynamic conditions (including accelerometer size effect separation) - See Reference 8, Reference 6 (or 9) Sections 7.1.1.1, 7.2.2.2, 7.3.3, or Reference 9 Section 19.1.8 for coning, sculling, scrolling definitions. These truth models (described in the Sections 2.1 and 2.2 to follow) are denoted as SPIN-CONE and SPIN-ROCK-SIZE.

Additional closed-form analytically exact truth models developed in Reference 6 (or 9) are SPIN-ACCEL (Sect. 11.2.2) for evaluating strapdown attitude update, acceleration transformation, velocity update algorithms under constant B Frame inertial angular-rate, constant B Frame specific-force-acceleration, constant N Frame inertial angular rate; and GEN NAV (Sect. 11.2.4) for evaluating strapdown attitude update, acceleration transformation, velocity/position update algorithms during long term navigation over an ellipsoidal earth surface shape model. The SPIN-ACCEL model can be easily expanded to also provide an analytically exact position solution.

Reference 6 (or 9) Section 11.2 shows how the previous defined analytical routines can be used to validate all subroutines typically utilized in a strapdown INS for attitude, velocity, position updating and associated system outputs.

Reference 6 (or 9) Section 11.1 also illustrates how specialized simulators can be designed for validating high speed strapdown integration algorithms that have been designed to identically match the equivalent true continuous integrals under particular angular-rate/specific-force-acceleration input conditions. This methodology is applied in Section 2.3 to follow for the Reference 10 coning, sculling, scrolling algorithms.

## 2.1 SPIN-CONE Truth Model

The SPIN-CONE truth model provides exact closed-form attitude and corresponding continuous integrated body frame angular rates for a spinning body with coning motion. The difference between integrated body rates at successive strapdown software sensor sampling cycles simulate the inputs from strapdown angular rate sensors used in the attitude update routines for the software under test. The SPIN-CONE and strapdown software computed attitude solutions are compared to establish strapdown software attitude algorithm accuracy.

The SPIN-CONE truth model is based on a closed-form solution to the attitude motion described by a body spinning at a fixed magnitude rotation rate and whose spin axis is rotating at a fixed precessional rate. The geometry of the motion is described in Figure 1 which shows the spin-axis and precessional-axis to be separated by an angle $\beta$. The spin axis rotates about the precessional axis which is defined to be perpendicular to a non-rotating inertial plane. A set of

body reference axes is implied in Figure 1 that rotates relative to a defined set of non-rotating coordinates.

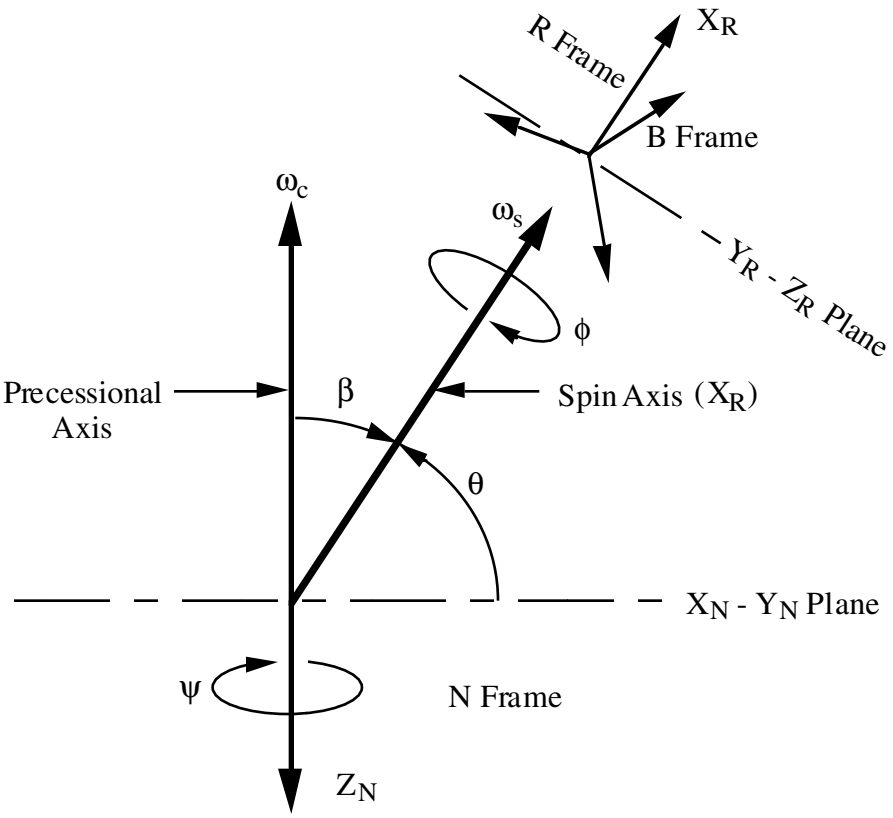


Figure 1 - SPIN-CONE Geometry

In Figure 1,

N = Non-rotating coordinate frame that is fixed to the non-rotating plane with $X_N$, $Y_N$ axes in the plane and the $Z_N$ axis perpendicular to the plane in the direction opposite the precessional rate vector.

R = Body "reference" coordinate axes fixed to the body with the X axis ($X_R$) along the spin axis. The R Frame is at a fixed orientation relative to B Frame sensor axes. A distinction is made between the B and R Frames so that the angular rate generated by the Figure 1 motion can have selected projections on B Frame sensor axes to test the general response of the strapdown attitude algorithms.

$\beta$ = Angle between the precessional axis and the R-Frame $X_R$ spin axis (the "cone angle") - considered constant.

$\omega_s$ = Inertial rotation rate of the body about $X_R$ ("spin rate") - considered constant.

$\omega_c$ = Inertial precessional rate of the body $X_R$ axis about the precessional axis which corresponds to a coning condition.

$\phi, \theta, \psi$ = Roll, pitch, heading Euler angles of the R Frame axes relative to the N Frame.

The analytical solution corresponding to the Figure 1 motion is (Ref. 6 (or 9) Sects. 11.2.1.1 and 11.2.1.2):

$$\phi = \left(\omega_s - \omega_c \cos\beta\right) t + \phi_0 \qquad \theta = \pi/2 - \beta \qquad \psi = -\omega_c t \qquad (1)$$

$$I\underline{\omega}^R_{IB}(t) \equiv \int_0^t \underline{\omega}^R_{IB}\, dt = \begin{bmatrix} \omega_s t \\ \dfrac{\omega_c \sin\beta}{\left(\omega_s - \omega_c \cos\beta\right)}\left(\cos\phi - \cos\phi_0\right) \\ -\dfrac{\omega_c \sin\beta}{\left(\omega_s - \omega_c \cos\beta\right)}\left(\sin\phi - \sin\phi_0\right) \end{bmatrix} \qquad (2)$$

$$I\underline{\omega}^B_{IB}(t) \equiv \int_0^t \underline{\omega}^B_{IB}\, dt = C^B_R\, I\underline{\omega}^R_{IB}(t) \qquad \Delta\underline{\alpha}_l \equiv \int_{t_{l-1}}^{t_l} \underline{\omega}^B_{IB}\, dt = I\underline{\omega}^B_{IB}(t_l) - I\underline{\omega}^B_{IB}(t_{l-1}) \qquad (3)$$

$$\begin{aligned} C_{RN_{11}} &= \cos\theta \cos\psi \\ C_{RN_{12}} &= -\cos\phi \sin\psi + \sin\phi \sin\theta \cos\psi \\ C_{RN_{13}} &= \sin\phi \sin\psi + \cos\phi \sin\theta \cos\psi \\ C_{RN_{21}} &= \cos\theta \sin\psi \\ C_{RN_{22}} &= \cos\phi \cos\psi + \sin\phi \sin\theta \sin\psi \\ C_{RN_{23}} &= -\sin\phi \cos\psi + \cos\phi \sin\theta \sin\psi \\ C_{RN_{31}} &= -\sin\theta \\ C_{RN_{32}} &= \sin\phi \cos\theta \\ C_{RN_{33}} &= \cos\phi \cos\theta \end{aligned} \qquad (4)$$

$$C^N_B = C^N_R\, C^R_B \qquad (5)$$

where

$\phi_0$ = Initial value for $\phi$. The initial value for $\psi$ is assumed to be zero.

t = Time from simulation start.

$l$ = Truth model output cycle time index corresponding to the highest speed computation repetition rate for the algorithms under test.

$\Delta\underline{\alpha}_l$ = Integrated B Frame $\underline{\omega}_{IB}$ inertial angular rate vector from cycle $l$-1 to $l$.

$C_{RN}(i,j)$ = Element in row i column j of $C_R^N$.

$C_B^R$ = Constant direction cosine matrix relating the B and R Frames.

The $\Delta\underline{\alpha}_l$ output vector would be used as the simulated angular rate sensor input to the attitude algorithms under test (e.g., Reference 10 Equations (8), (12) and (24) with zero setting for the N Frame rotation rate and *l* corresponding to the high speed coning algorithm computation cycle index). The $C_B^N$ matrix represents the truth solution corresponding to the $\Delta\underline{\alpha}_l$ history for comparison with the equivalent $C_B^N$ generated by the algorithms under test. Comparison is performed by multiplying the algorithm computed $C_B^N$ (on the left) by the transpose of the truth model $C_B^N$ (on the right) and comparing the result with the identity matrix (the correct value of the product when the algorithm computed $C_B^N$ is error free) - See Reference 6 (or 9) Section 11.2.1.4 for details and how results can be equated to equivalent normality, orthogonality and misalignment errors.

If the algorithms being tested are exact and properly programmed, the comparison described previously with the SPIN-CONE truth solution should show identically zero error. The attitude algorithms in Reference 10 Equations (8) with (12) are exact under zero N Frame rotation rate. An exact comparison with SPIN-CONE should be obtained when using zero coning rate (i.e., by setting $\omega_c$ to zero and the coning term in Reference 10 Equations (12) to zero). With non-zero $\omega_c$, (and the Reference 10 Equations (12) coning term active in the algorithms being tested) the comparison with SPIN-CONE measures the error in the coning computation portion of the algorithms (a function of the *l* cycle rate). If the coning computation algorithm is an analytically exact solution to an assumed form of the angular rate input profile (e.g., Ref. 10 Eqs. (24)), Section 2.3 to follow shows how the associated coning algorithm software can also be exactly validated (i.e., with zero error).

## 2.2 SPIN-ROCK-SIZE Truth Model

The SPIN-ROCK-SIZE truth model provides exact closed form integrated angular rates, integrated linear accelerations, attitude, velocity and position simulating a strapdown sensor assembly undergoing spinning/sculling/scrolling dynamic motion with the individual accelerometers mounted at specified lever arm locations within the sensor assembly (i.e., simulating size effect separation). The integrated rates and accelerations are used as inputs to strapdown software algorithms under test to compute body attitude, accelerometer size effect lever arm compensation to the body navigation reference center, transformation of compensated specific force acceleration to navigation coordinats, and transformed acceleration integration to velocity and position. The strapdown software algorithm accuracy is evaluated by comparing the SPIN-ROCK-SIZE truth model computed position, velocity and attitude with the equivalent data generated by the strapdown software algorithms under test.

The SPIN-ROCK-SIZE truth model generates navigation and inertial sensor outputs under dynamic motion around an arbitrarily specified and fixed rotation axis (Figure 2). The rotation axis is defined to be non-rotating and non-accelerating. The dynamic motion is characterized as rigid body motion around the specified axis with the specified axis located within the rotating rigid body. The strapdown sensor assembly being simulated is located in the rigid body and has its navigation reference center at a specified lever arm location from the rotation axis. Each accelerometer within the sensor assembly is located at an arbitrarily selected lever arm position. The accelerations measured by the accelerometers are created by centripetal and tangential acceleration effects produced by their lever arm displacement from the rotation axis under rigid body dynamic angular motion around the rotation axis. For this truth model, the N Frame is inertially non-rotating and gravity is zero.

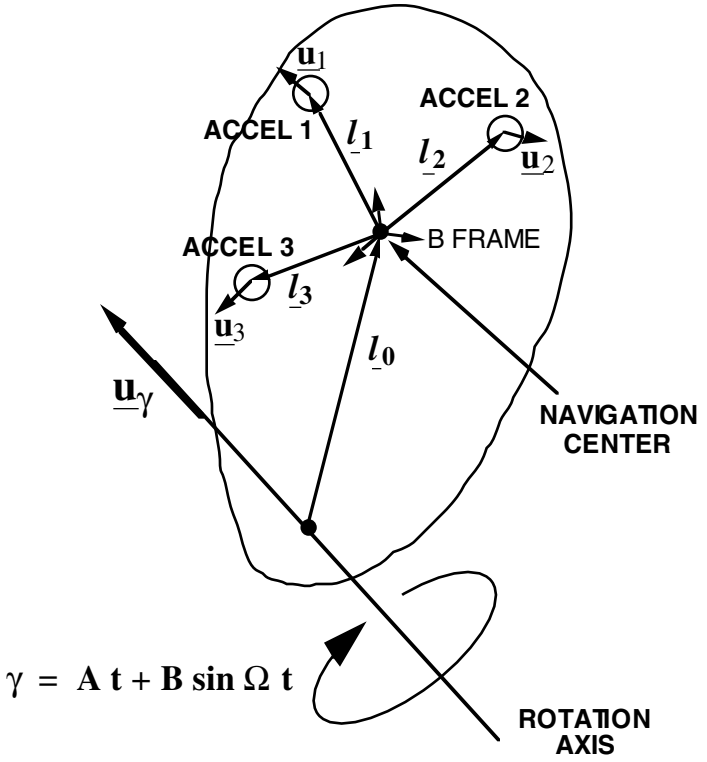


**Figure 2 - SPIN-ROCK-SIZE Parameters**

In Figure 2,

$\underline{l}_0$ = Position vector from the rotation axis to the navigation center.

$\underline{l}_i$ = Position vector from the navigation center to the accelerometer i (Accel i) center of seismic mass.

$\underline{u}_i$ = Accelerometer i input axis.

$\underline{u}_\gamma$ = Unit vector along the angular rotation axis.

$\gamma$ = Angle of rotation about $\underline{u}_\gamma$.

A, B, $\Omega$ = Constants.

The analytical solution corresponding to the Figure 2 motion is (Ref. 6 (or 9) Sects. 11.2.3.1 - 11.2.3.3):

$$\gamma = A\,t + B \sin \Omega t \qquad \dot{\gamma} = A + B\,\Omega \cos \Omega t \qquad (6)$$

$$\Delta\underline{\alpha}_l = \int_{t_{l-1}}^{t_l} \underline{\omega}_{IB}^{B}\, dt = \left(\gamma(t_l) - \gamma(t_{l-1})\right) \underline{u}_\gamma^B \qquad (7)$$

$$\Delta v_{i_l} = \int_{t_{l-1}}^{t_l} \underline{u}_i^B \cdot \underline{a}_{SF_i}^B\, dt = \underline{u}_i^B \cdot \left\{ \left[ \left(f_a(t_l) - f_a(t_{l-1})\right) \left(\underline{u}_\gamma^B \times\right) + \left(f_b(t_l) - f_b(t_{l-1})\right) \left(\underline{u}_\gamma^B \times\right)^2 \right] \left(\underline{l}_0^B + \underline{l}_i^B\right) \right\} \qquad (8)$$

$$f_a(t) = B\,\Omega \cos \Omega t \qquad f_b(t) = \left(A^2 + \frac{1}{2} B^2\,\Omega^2\right) t + 2\,A\,B \sin \Omega t + \frac{1}{2} B^2\,\Omega \sin \Omega t \cos \Omega t \qquad (9)$$

$$C_B^N = C_{B_0}^N\, C_B^{B_0} \qquad C_B^{B_0} = I + \sin\gamma \left(\underline{u}_\gamma^B \times\right) + \left(1 - \cos\gamma\right) \left(\underline{u}_\gamma^B \times\right)^2 \qquad (10)$$

$$\underline{v}^N = \dot{\gamma}\, C_B^N \left(\underline{u}_\gamma^B \times \underline{l}_0^B\right) \qquad \underline{R}^N = C_B^N\, \underline{l}_0^B \qquad (11)$$

where

$I$ = Identity matrix.

$C_{B_0}^N$ = Initial value of $C_B^N$.

$\underline{a}_{SF_i}$ = Specific force acceleration vector at the accelerometer i location. Specific force acceleration is 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 $\underline{a}_{SF}$.

$\Delta v_{i_l}$ = Integrated specific force acceleration along the accelerometer i input axis over the computation algorithm high speed $l$ cycle time interval from $l$-1 to $l$.

The $\Delta\underline{\alpha}_l$, $\Delta v_{i_l}$ output vectors would be used as the simulated angular rate sensor and accelerometer inputs to the attitude update, acceleration transformation, velocity update, position update, size effect compensation algorithms under test (e.g., Reference 10 Equations (8) - (10, (12) - (17), (35), 37) and (42) - (43) with zero setting for the N Frame inertial rotation rate and $l$ corresponding to the high speed coning/sculling/scrolling algorithm computation cycle index)

The $C_B^N$ matrix represents the attitude truth solution corresponding to the $\Delta\underline{\alpha}_l$ history for comparison with the equivalent $C_B^N$ generated by the algorithms under test. Comparison is performed as described in Section 2.1. The $\underline{v}^N$ vector is the velocity truth solution used for comparison against the equivalent $\underline{v}^N$ generated by integration using the algorithms under test. The $\underline{R}^N$ vector is the truth model position solution used for comparison against the equivalent $\underline{R}^N$ generated by integration using the algorithms under test (e.g., summation of the $\Delta\underline{R}_m^N$ increments in Equations (10) of Reference 10).

If the algorithms being tested are exact and properly programmed, the comparison described previously with the SPIN-ROCK-SIZE truth solution should show identically zero error. The attitude algorithms in Reference 10 Equations (8) with (12) are exact under zero N Frame rotation rate. Hence, since SPIN-ROCK-SIZE is based on constant angular rate vector direction (i.e., zero coning), an exact comparison with the SPIN-ROCK-SIZE attitude solution should be obtained when setting the coning term in the Reference 10, Equations (12) rotation vector calculation to zero. The acceleration-transformation/velocity-update/ position-update algorithms in Reference 10 Equations (9) - (10) and (12) are exact under zero N Frame rotation rate, hence, are also exact under the simpler restriction of constant B Frame angular rate and specific force acceleration. Constant B Frame angular-rate/specific-force can be generated with SPIN-ROCK-SIZE by setting the B coefficient to zero. Under this condition and zero accelerometer lever arms, an exact comparison of the previous algorithms with the SPIN-ROCK-SIZE attitude/velocity/position solution should be obtained. With non-zero B coefficient and simulated accelerometer lever arms included, the comparison with SPIN-ROCK-SIZE measures the error in sculling/scrolling and accelerometer size effect compensation elements of the algorithms being tested. For the previous example, sculling/scrolling/size-effect compensation calculations can be added to the test by activating the Reference 10 Equations (24), (25), (35), (37) and (42) - (43) to go with the Equations (8) - (10) and (12) attitude/velocity/position update algorithms.

If the sculling and scrolling computation algorithms are analytically exact solutions to an assumed form of the angular-rate/specific-force-acceleration input profile (e.g., Ref. 10 Eqs. (25) and (26)), Section 2.3 to follow shows how the associated sculling/scrolling algorithm software can be exactly validated (i.e., with zero error).

### 2.3 Specialized Simulators For High Speed Algorithm Validation

High speed strapdown inertial digital integration algorithms designed to be exact under assumed analytic forms of their inertial sensor inputs can be validated numerically using specialized simulators. The general methodology is described in Reference 6 (or 9) Section 11.1. For example, consider the strapdown inertial high speed coning, sculling, scrolling integration functions in Section 3.4 Equations (13) - (17):

$$\underline{\alpha}(t) = \int_{t_{m-1}}^{t} \underline{\omega}_{IB}^{B}\, d\tau \qquad \underline{\upsilon}(t) = \int_{t_{m-1}}^{t} \underline{a}_{SF}^{B}\, d\tau$$

$$\underline{S}_{\upsilon}(t) = \int_{t_{m-1}}^{t} \underline{\upsilon}(\tau)\, d\tau \qquad \underline{S}_{\upsilon_m} = \underline{S}_{\upsilon}(t_m)$$

$$\Delta\underline{\phi}_{Cone_m} = \frac{1}{2}\int_{t_{m-1}}^{t_m} \left(\underline{\alpha}(t) \times \underline{\omega}_{IB}^{B}\right) dt \tag{12}$$

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

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

where

m = Navigation parameter (i.e., attitude, velocity, position) update cycle time index.

$\underline{\omega}_{IB}$ = Inertial angular rate vector that would be measured by the strapdown angular rate sensors.

$\underline{\alpha}$ = Integrated inertial angular rate.

$\underline{a}_{SF}$ = Specific force acceleration vector that would be measured by the strapdown accelerometers.

$\underline{\upsilon}$ = Integrated specific force acceleration.

$\underline{S}_{\upsilon}$ = Doubly integrated specific force acceleration.

$\Delta\underline{\phi}_{Cone_m}$ = Coning contribution to rotation vector from cycle time m-1 to m.

$\Delta\underline{\eta}_{Scul_m}$ = Sculling contribution to velocity translation vector from cycle time m-1 to m.

$\Delta\underline{\kappa}_{Scrl_m}$ = Scrolling contribution to position translation vector from cycle time m-1 to m.

In Reference 6 (or 9) Sections 7.1.1.1.1, 7.2.2.2.2 and Reference 10 Section 4.3, digital integration algorithms are designed to implement the previous operations using a high speed *l* cycle computation rate between attitude, velocity, position m cycle updates. The algorithms (Reference 10 Equations (24) - (26)) are designed to provide exact solutions to the above operations under linearly ramping angular rate and specific force acceleration profiles between *l* cycles. Algorithm inputs are integrated angular rate and specific force acceleration increments

between $l$ cycles, representing the input signals from strapdown angular rate sensors and accelerometers. A simple method for numerically validating that the algorithms perform as designed is to build a specialized simulator that generates integrated inertial sensor increment inputs to the algorithms based on a linear ramping angular-rate/specific-force-acceleration profile. The algorithms to be validated would then be operated in the simulation at their $l$ cycle rate using the simulated sensor incremental inputs, and evaluated at the m cycle times. For correctly derived and software implemented algorithms, results should exactly match the true analytic integral of Equations (12) under linear ramping angular-rate/specific-force-acceleration conditions:

$$\underline{\omega}^{B}_{IB} = \underline{A}_0 + \underline{A}_1 (t - t_{m-1}) \qquad \underline{a}^{B}_{SF} = \underline{B}_0 + \underline{B}_1 (t - t_{m-1}) \qquad (13)$$

where

$\underline{A}_0, \underline{A}_1, \underline{B}_0, \underline{B}_1$ = Selected simulation constants.

Substituting Equations (13) into (12) and carrying out the integral operations analytically yields the true analytic solutions corresponding to the assumed linear ramping profiles:

$$\Delta\underline{\phi}_{Cone_m} = \frac{1}{12}\left(\underline{A}_0 \times \underline{A}_1\right) T_m^3 \qquad \Delta\underline{\eta}_{Scul_m} = \frac{1}{12}\left(\underline{A}_0 \times \underline{B}_1 + \underline{B}_0 \times \underline{A}_1\right) T_m^3$$

$$\Delta\underline{\kappa}_{Scrl_m} = \frac{1}{72}\left(2\,\underline{A}_0 \times \underline{B}_1 - 3\,\underline{A}_1 \times \underline{B}_0\right) T_m^4 - \frac{1}{360}\left(\underline{A}_1 \times \underline{B}_1\right) T_m^5 \qquad (14)$$

$$\underline{S}_{\upsilon_m} = \frac{1}{2}\underline{B}_0 T_m^2 + \frac{1}{6}\underline{B}_1 T_m^3$$

where

$T_m$ = Time interval between computation m cycles.

The $l$ cycle incremental inputs to the algorithms being validated are the integrals of Equations (13) between $l$ cycles:

$$\Delta\underline{\alpha}_l = \int_{t_{l-1}}^{t_l} \underline{\omega}^{B}_{IB}\, dt = \underline{A}_0 T_l + \frac{1}{2}\underline{A}_1 \left((t_l - t_{m-1})^2 - (t_{l-1} - t_{m-1})^2\right)$$

$$\Delta\underline{\upsilon}_l = \int_{t_{l-1}}^{t_l} \underline{a}^{B}_{SF}\, dt = \underline{B}_0 T_l + \frac{1}{2}\underline{B}_1 \left((t_l - t_{m-1})^2 - (t_{l-1} - t_{m-1})^2\right) \qquad (15)$$

where

$l$ = High speed algorithm computation cycle index (within the m update cycle).

$T_l$ = Time interval between $l$ cycles.

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

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

Operating the Reference 10 Equation (24) - (26) high speed digital integration algorithms with Equation (15) inputs should provide results at the m cycle times that identically match Equations (14) for any values selected for the $\underline{A}_0$, $\underline{A}_1$, $\underline{B}_0$, $\underline{B}_1$ constants.

## 3. VIBRATION EFFECTS ANALYSIS

Strapdown inertial navigation integration algorithms are designed to accurately account for three-dimensional high frequency angular and linear vibration of the sensor assembly. If not properly accounted for, such motion can lead to systematic attitude/velocity/position error build-up. The high speed algorithms described in Reference 10 Equations (24) - (26) to measure these effects (i.e., coning, sculling, scrolling, doubly integrated sensor input) are based on approximations to the form of the angular-rate/specific-force profiles during the high speed update interval. An important part of the algorithm design is their accuracy evaluation under hypothesized vibration exposure of the strapdown INS in the user vehicle, the subject of this section. Algorithm performance evaluation results, used in design/synthesis iterative fashion, eventually set the order of the algorithm selected and its required repetition rate in the INS computer.

Since the sensor assembly is dynamically coupled to the INS mount through the INS structure (in many cases including mechanical isolators and their imbalances), vibrations input to the INS mount become dynamically distorted as they translate into inertial sensor outputs provided to the navigation algorithms. Included in this section is a description of a simplified analytical model for characterizing the dynamic response of an INS sensor assembly to input vibration and its use in system performance evaluation.

All equations in this section are written in B Frame coordinates whose explicit designation has been deleted for analytical simplicity.

### 3.1 System Response Under Sinusoidal Vibration

In this section we describe the effect of sensor assembly linear and angular sinusoidal vibration on system navigational performance. The section is divided into two major subsections covering true attitude, velocity, position motion vibration response, and the vibration response of particular algorithms used in the system attitude, velocity, position digital integration routines. The material is selected from Section 10.1 (and its subsections) of Reference 6 (or 9) which also covers other vibration induced effects.

The attitude response discussion is based on the following B Frame input angular vibration designed to produce coning motion:

$$\underline{\theta}(t) = \underline{u}_x \theta_{0_x} \sin\left(\Omega t - \varphi_{\theta_x}\right) + \underline{u}_y \theta_{0_y} \sin\left(\Omega t - \varphi_{\theta_y}\right) \qquad (16)$$

where

$\underline{\theta}(t)$ = B Frame vibration "angle" vector defined as the integrated B Frame inertial angular rate. Since we are addressing angular vibration effects that are by nature, small in amplitude, $\underline{\theta}(t)$ is approximately the rotation vector associated with the vibration motion, hence, represents an actual physical angle vector (See Reference 6 (or 9) Section. 3.2.2 for rotation vector definition).

$\underline{u}_x, \underline{u}_y$ = Unit vectors along the B Frame X, Y axes.

$\Omega$ = Vibration frequency.

$\theta_{0_x}, \theta_{0_y}$ = Sinusoidal vibration "angle" vector amplitude around B Frame axes X and Y.

$\varphi_{\theta_x}, \varphi_{\theta_y}$ = Phase angle associated with each B Frame X, Y axis angular vibration.

The velocity response discussion is based on the following B Frame input linear and angular vibration designed to produce sculling motion:

$$\underline{\theta}(t) = \underline{u}_x \theta_{0_x} \sin\left(\Omega t - \varphi_{\theta_x}\right) \qquad \underline{a}_{SF}(t) = \underline{u}_y a_{SF0_y} \sin\left(\Omega t - \varphi_{a_{SF_y}}\right) \qquad (17)$$

where

$a_{SF0_y}$ = Sinusoidal vibration amplitude of the B Frame Y axis specific force acceleration vibration.

$\varphi_{a_{SF_y}}$ = Phase angle associated with the B Frame Y axis linear vibration.

Note that because the angular motion is about a fixed axis, there is no coning motion in the previous vibration profile.

The position response discussion is based on B Frame linear vibration which can produce folding effect amplification in the position update algorithms. Such effects are generally not present in the attitude/velocity algorithms because the inertial sensors are typically of the integrating type, providing their inputs to the navigation computer in the form of pre-integrated angular rate and specific force acceleration increments. The B Frame input vibration is as follows:

$$\underline{a}_{SF}(t) = \underline{u}_{Vib} a_{SF_0} \sin\left(\Omega t - \varphi_{a_{SF}}\right) \qquad \underline{\theta}(t) = 0 \qquad (18)$$

where

$\underline{u}_{Vib}$ = Linear vibration input axis.

Note that because there is no angular motion in the previous vibration profile, there is no coning, sculling or scrolling effect on the resulting position response.

### 3.1.1 True System Response

Under the Equation (16) vibration profile, the following true attitude motion is generated (Ref. 6 (or 9) Sect. 10.1.1.1):

$$\begin{aligned}\underline{\Phi}(t) = {} & \underline{u}_x \theta_{0_x}\left[\sin\left(\Omega t - \varphi_{\theta_x}\right) - \sin\left(\Omega t_0 - \varphi_{\theta_x}\right)\right] \\ & + \underline{u}_y \theta_{0_y}\left[\sin\left(\Omega t - \varphi_{\theta_y}\right) - \sin\left(\Omega t_0 - \varphi_{\theta_y}\right)\right] \\ & + \underline{u}_z \frac{1}{2}\Omega \theta_{0_x} \theta_{0_y} \sin\left(\varphi_{\theta_y} - \varphi_{\theta_x}\right)\left[(t - t_0) - \frac{\sin\left(\Omega (t - t_0)\right)}{\Omega}\right]\end{aligned} \qquad (19)$$

where

$t_0$ = Initial time t.

$\underline{\Phi}(t)$ = Rotation vector describing the B Frame attitude at time t due to the Equation (16) vibration, relative to the B Frame attitude at $t_0$.

The attitude response has first order constant and oscillatory terms around the angular vibration input axes, a second order angular vibration around $\underline{u}_z$ (the axis perpendicular to the angular vibration input axes), and a linear time build-up term around axis $\underline{u}_z$ representing the coning effect. The average slope of the attitude response is the linear term coefficient denoted as the coning rate (previous reference):

$$\dot{\underline{\Phi}}_{Avg} = \underline{u}_z \frac{1}{2}\Omega \theta_{0_x} \theta_{0_y} \sin\left(\varphi_{\theta_y} - \varphi_{\theta_x}\right) = \text{Coning rate} \qquad (20)$$

Under the Equation (17) vibration profile, the following true velocity motion is generated (Ref. 6 (or 9) Sect. 10.1.2.1):

$$\begin{aligned}\underline{v}(t) = {} & \underline{u}_y a_{SF0_y} \frac{1}{\Omega}\left[\cos\left(\Omega t_0 - \varphi_{a_{SF_y}}\right) - \cos\left(\Omega t - \varphi_{a_{SF_y}}\right)\right] \\ & + \underline{u}_z \frac{1}{2}\theta_{0_x} a_{SF0_y}\left\{\frac{1}{\Omega}\left[\sin\left(\Omega t - \varphi_{\theta_x}\right) - \sin\left(\Omega t_0 - \varphi_{\theta_x}\right)\right]\left[\cos\left(\Omega t_0 - \varphi_{a_{SF_y}}\right)\right.\right. \\ & \left.\left. - \cos\left(\Omega t - \varphi_{a_{SF_y}}\right)\right] + \cos\left(\varphi_{a_{SF_y}} - \varphi_{\theta_x}\right)\left[(t - t_0) - \frac{\sin\left(\Omega (t - t_0)\right)}{\Omega}\right]\right\}\end{aligned} \qquad (21)$$

where

$\underline{v}(t)$ = Velocity at time t in the time $t_0$ oriented B Frame due to the Equation (17) angular/linear vibration since time $t_0$.

The velocity response has first order constant and oscillatory terms along the linear vibration input axis, second order constant and oscillatory terms along $\underline{u}_z$ (the axis perpendicular to the linear/angular vibration input axes), and a linear time build-up term along axis $\underline{u}_z$ representing the sculling effect. The average slope of the velocity response is the linear term coefficient denoted as the sculling rate (previous reference):

$$\dot{\underline{v}}_{Avg} = \underline{u}_z \frac{1}{2} \theta_{0_x} a_{SF0_y} \cos\left(\varphi_{a_{SF_y}} - \varphi_{\theta_x}\right) = \text{Sculling Rate} \qquad (22)$$

Under the Equation (18) vibration profile, the following true velocity, position motion is generated (Ref. 6 (or 9) Sect. 10.1.3.2.1):

$$\underline{v}(t) = \int_{t_0}^{t} \underline{a}_{SF}(\tau)\, d\tau = - \underline{u}_{Vib}\, a_{SF_0} \frac{1}{\Omega} \left[\cos(\Omega t_0 - \varphi_{a_{SF}}) - \cos(\Omega t - \varphi_{a_{SF}})\right] \qquad (23)$$

$$\underline{R}(t) = \int_{t_0}^{t} \underline{v}(\tau)\, d\tau = - \underline{u}_{Vib}\, a_{SF_0} \frac{1}{\Omega} \left\{(t - t_0) \cos(\Omega t_0 - \varphi_{a_{SF}}) - \frac{1}{\Omega}\left[\sin(\Omega t - \varphi_{a_{SF}}) - \sin(\Omega t_0 - \varphi_{a_{SF}})\right]\right\} \qquad (24)$$

where

$\underline{R}(t)$ = Position at time t in the time $t_0$ oriented B Frame due to Equation (18) vibration since time $t_0$.

### 3.1.2 System Algorithm Response

The response of the system attitude, velocity, position computational algorithms to the Section 3.1 input vibrations depends on the particular algorithms utilized. An important part of algorithm design is an analytical assessment of their response in comparison with the true kinematic response under hypothesized input motion. For the two-speed algorithms described in Reference 10, the low speed portions have been designed to be analytically exact such that algorithm errors are generated only by the high speed algorithms (except for minor small trapezoidal integration algorithm errors associated with Coriolis, gravity, N Frame rotation rate terms). The result is that under the Section 3.1 input profiles, the Reference 10 algorithm response should match the Section 3.1 truth solution plus an added high speed algorithm err

For the Reference 10 (and 4) attitude computation, a high speed algorithm computes the coning contribution to the rotation vector (Ref. 10 Eqs. (24)) based on a second order truncated Taylor series expansion as:

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

$$\underline{\alpha}_l = \sum_l \Delta\underline{\alpha}_l \quad \text{From } t_{m-1} \text{ to } t_l \tag{25}$$

For the previous coning algorithm operating with an exact attitude updating algorithm (Ref. 4 and Ref. 10 Eqs. (8)), the average algorithm error response under the Equations (16) vibration profile is (Ref. 6 (or 9) Sect. 10.1.1.2.2):

$$\delta\dot{\underline{\Phi}}_{\text{Algo}} = \delta\Delta\dot{\underline{\phi}}_{\text{ConeAlgo}} = \underline{u}_z \frac{1}{2}\,\Omega\,\theta_{0_x}\,\theta_{0_y}\sin\left(\varphi_{\theta_y} - \varphi_{\theta_x}\right)\left\{\left[1 + \frac{1}{3}\left(1 - \cos\Omega T_l\right)\right]\left(\frac{\sin\Omega T_l}{\Omega T_l}\right) - 1\right\} \tag{26}$$

where

$\delta\dot{\underline{\Phi}}_{\text{Algo}}$, $\delta\Delta\dot{\underline{\phi}}_{\text{ConeAlgo}}$ = Average attitude and coning algorithm error rate.

For the Reference 10 (and 5) velocity computation, a high speed algorithm computes the sculling contribution to the velocity translation vector (Ref. 10 Eqs. (25)) based on a second order truncated Taylor series expansion as:

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

$$\underline{\upsilon}_l = \sum_l \Delta\underline{\upsilon}_l \quad \text{From } t_{m-1} \text{ to } t_l \tag{27}$$

For the previous sculling algorithm operating with an exact velocity updating algorithm (Ref. 5 and Ref. 10 Eqs. (9)), the average algorithm error response under the Equations (17) vibration profile is (Ref. 6 (or 9) Sect. 10.1.2.2.2):

$$\dot{\underline{\delta v}}_{\text{Algo}} = \delta\dot{\underline{\Delta\eta}}_{\text{ScullAlgo}} = \underline{u}_z \frac{1}{2}\,\theta_{0_x}\,a_{SF0_y}\cos\left(\varphi_{a_{SF_y}} - \varphi_{\theta_x}\right)\left\{\left[1 + \frac{1}{3}\left(1 - \cos\Omega T_l\right)\right]\left(\frac{\sin\Omega T_l}{\Omega T_l}\right) - 1\right\} \tag{28}$$

where

$\dot{\underline{\delta v}}_{\text{Algo}}$, $\delta\dot{\underline{\Delta\eta}}_{\text{ScullAlgo}}$ = Average velocity update and sculling algorithm error rate.

Because there is no coning motion in the Equations (17) vibration profile, the accompanying Reference 10 attitude algorithm response would be error free.

The Reference 10 (and 5) position translation vector computation uses a high speed algorithm to compute doubly integrated acceleration (Ref. 10 Eqs. (26)) based on a second order truncated Taylor series expansion as:

$$\underline{S}_{\upsilon_m} = \int_{t_{m-1}}^{t_m}\int_{t_{m-1}}^{\tau} \underline{a}_{SF}^{B}\, d\tau_1\, d\tau \approx \sum_l \left[\underline{\upsilon}_{l\text{-}1}T_l + \frac{T_l}{12}\left(5\,\Delta\underline{\upsilon}_l + \Delta\underline{\upsilon}_{l\text{-}1}\right)\right] \quad \text{From } t_{m-1} \text{ to } t_m \tag{29}$$

where

$\underline{\upsilon}_l$ = As defined previously in Equations (27).

For the previous doubly integrated acceleration algorithm operating with an exact position updating algorithm (Ref. 5 or Ref. 10 Eqs. (10) with (12)), the position error response under the Equations (18) vibration profile is (Ref. 6 (or 9) Sects. 10.1.3.2.3):

$$\begin{aligned}\delta\underline{R}_{Algo}(t) = \sum_m \delta\underline{S}_{\upsilon_m} = -\underline{u}_{Vib}\frac{1}{\Omega^2}a_{SF_0}\left\{\left(\frac{\Omega T_l \sin \Omega' T_l}{2(1-\cos \Omega' T_l)} - 1 + \frac{1}{12}\Omega T_l \sin \Omega' T_l\right)\left[\sin\left(\Omega'(t-t_0) + \Omega t_0 - \varphi_{a_{SF}}\right) - \sin(\Omega t_0 - \varphi_{a_{SF}})\right]\right. \\ \left. - \frac{1}{12}\Omega T_l\left[\cos\left(\Omega'(t-t_0) + \Omega t_0 - \varphi_{a_{SF}}\right) - \cos(\Omega t_0 - \varphi_{a_{SF}})\right](1-\cos \Omega' T_l)\right\}\end{aligned} \tag{30}$$

$$\frac{\Omega' T_l}{2\pi} = \frac{\Omega T_l}{2\pi} - \left(\frac{\Omega T_l}{2\pi}\right)_{Intgr} \qquad k = \left(\frac{\Omega T_l}{2\pi}\right)_{Intgr} \qquad \Omega' \equiv \Omega - \frac{2\pi k}{T_l}$$

where

$\delta\underline{R}_{Algo}(t)$ = Position algorithm error.

$\delta\underline{S}_{\upsilon_m}$ = Error in the $\underline{S}_{\upsilon_m}$ acceleration double integration algorithm.

k = Nearest integer value of the ratio of $\Omega$ to $2\pi / T_l$.

$( )_{Intg}$ = ( ) rounded to the nearest integer value (e.g., $(0.3)_{Intgr} = 0$, $(0.5)_{Intgr} = 1$, $(0.7)_{Intgr} = 1$, $(1.3)_{Intgr} = 1$, $(1.5)_{Intgr} = 2$, $(1.7)_{Intgr} = 2$, etc.).

$\Omega'$ = Folded frequency.

Because there is no coning or sculling motion in the Equations (18) vibration profile, the accompanying Reference 10 attitude and velocity algorithm response would be error free.

Equations (30) show that the algorithm computed position error can be sizable when the folded frequency $\Omega'$ approaches zero (i.e., when $\Omega$ is close to an integer multiple of $2\pi / T_l$ for which $(1 - \cos \Omega' T_l)$ approaches zero). Reference 6 (or 9) Section 10.1.3.2.3 shows that for k = 0, the term of concern $\frac{\Omega T_l \sin \Omega' T_l}{2(1-\cos \Omega' T_l)} = 1$ but for k > 0, $\frac{\Omega T_l \sin \Omega' T_l}{2(1-\cos \Omega' T_l)}$ equals $\frac{2\pi k}{\Omega' T_l}$ which is infinite for zero folding frequency $\Omega'$. The latter effect on position error is actually a build-up in time that only becomes infinite at infinite time (previous reference). To assess the effect for finite time, the equivalent to Equations (30) is (Ref. 6 (or 9) Sect. 10.1.3.2.4):

$$\delta \underline{R}_{Algo}(t) = -\underline{u}_{Vib} \frac{1}{\Omega^2} a_{SF_0} \left\{ \Omega(t - t_0) \left( \frac{f_1(\Omega' T_l)}{2 f_2(\Omega' T_l)} - \frac{\Omega'}{\Omega} \right.\right.$$
$$\left. + \frac{1}{12} (\Omega' T_l)^2 f_1(\Omega' T_l) \right) \Big[ \cos(\Omega t_0 - \varphi_{a_{SF}}) f_1\big(\Omega'(t - t_0)\big) \qquad (31)$$
$$- \sin(\Omega t_0 - \varphi_{a_{SF}}) \Omega'(t - t_0) f_2\big(\Omega'(t - t_0)\big) \Big]$$
$$\left. - \frac{1}{12} \Omega T_l \Big[ \cos\big(\Omega'(t - t_0) + \Omega t_0 - \varphi_{a_{SF}}\big) - \cos(\Omega t_0 - \varphi_{a_{SF}}) \Big] (1 - \cos \Omega' T_l) \right\}$$

in which the $f_1$, $f_2$ functions are defined as:

$$f_1(x) \equiv \frac{\sin x}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \cdots \qquad f_2(x) \equiv \frac{(1 - \cos x)}{x^2} = \frac{1}{2!} - \frac{x^2}{4!} + \frac{x^4}{6!} - \cdots \qquad (32)$$

Equation (31) for the position algorithm error is singularity free for finite values of time t and for all values of $\Omega'$ (i.e., including $k > 0$ values).

### 3.2 System Vibration Analysis Model

The results of Section 3.1 are based on having knowledge of the INS sensor assembly B Frame vibration input amplitudes and phasing that are representative of expected system usage. Finding values for these terms can be a time consuming computer aided software design process involving complex mechanical modeling of the INS structure and how it mechanically couples to a user vehicle. Due to its complexity, the process is inherently prone to data input error that distorts results obtained. To provide a reasonableness check on the results, simplified dynamic models are frequently employed for comparison that lend themselves to closed-form analytical solutions. Once the detailed modeling results match the simplified model within its approximation uncertainty, the detailed model is deemed valid for use in estimating B Frame response.

From a broader perspective, it must be recognized that it is virtually impossible to develop an accurate mechanical dynamic model for an INS in a user vehicle due to variations in mechanical structural properties between INSs of a particular design (e.g., variations in stiffness/damping characteristics of electronic circuit boards in their respective card guides, variations in mechanical housings, variations in mounting interfaces, etc.), as well as variations in the characteristics for a particular INS over temperature and time. On the other hand, for performance analysis purposes, only "ball-park" accuracy is generally required for B Frame vibration characteristics. All things considered, it becomes reasonable to use the simplified analytical models for B Frame vibration, thereby eliminating the need for cumbersome computerized modeling.

Figure 3 illustrates such a simplified analytical model depicting the INS sensor assembly linear and angular response to linear INS input vibration exposure.

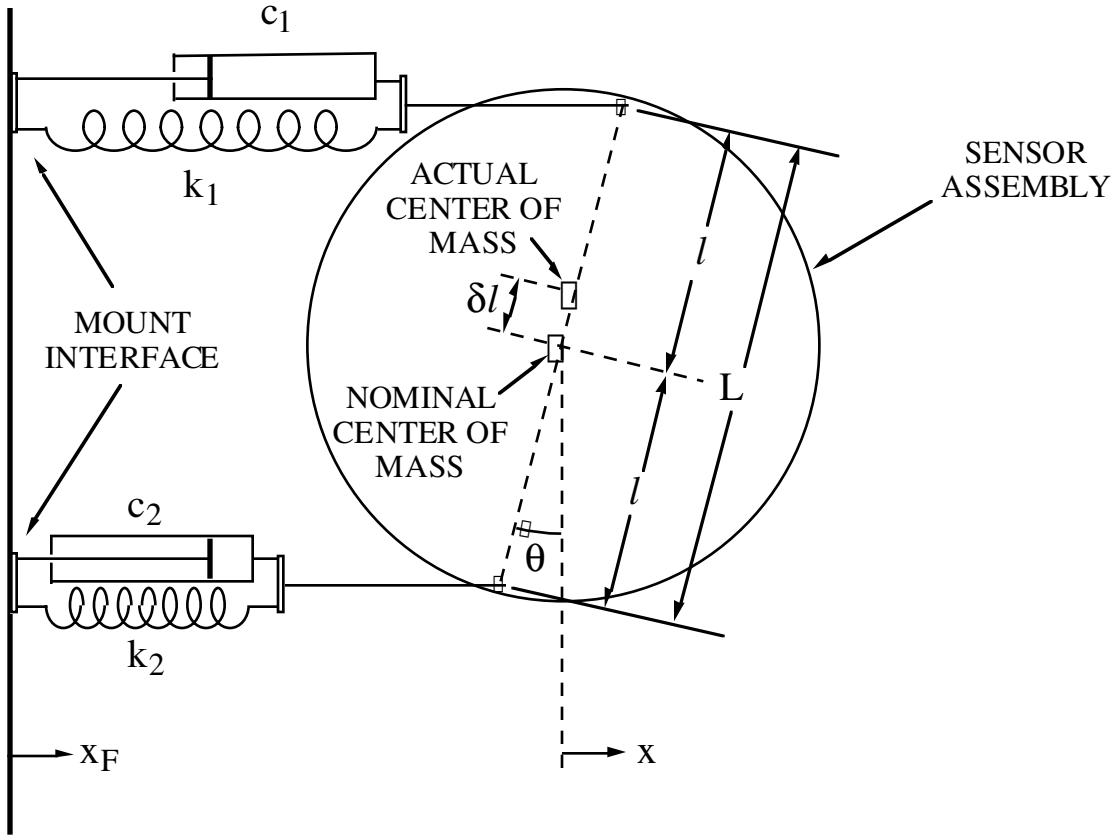


**Figure 3 - Simplified Sensor Assembly Dynamic Response Model**

In Figure 3

$X_F$, X = Vibration forcing function input position displacement and sensor assembly position response.

$\theta$ = Sensor assembly angular response to $X_F$ input vibration.

$k_i$, $c_i$ = Spring constants and damping coefficients for structure connecting the sensor assembly to the INS vibration input source.

$\delta l$ = Variation of the actual sensor assembly center of mass from its nominal location.

Figure 3 depicts a sensor assembly that would be nominally mounted with a symmetrical attachment to the vibration source such that $k_1$, $c_1$ and $k_2$, $c_2$ are nominally equal with the actual sensor assembly center of mass collocated with the nominal center of mass (zero $\delta l$). Under such nominal "CG Mount" conditions, the input vibration $X_F$ produces sensor assembly motion with zero angular response $\theta$ and with a linear X response of (Ref. 6 (or 9) Sect. 10.5.1):

$$A(S) = \frac{(2\,c\,S + 2\,k)}{m\,S^2 + 2\,c\,S + 2\,k}\,A_F(S) \qquad (33)$$

where

$A_F(S)$, $A(S)$ = Laplace transforms of the input vibration and sensor assembly response accelerations (the second derivatives of $X_F$, $X$).

$S$ = Laplace transform variable.

$k$, $c$ = Nominal values for $k_i$, $c_i$.

$m$ = Sensor assembly mass.

Under off-nominal conditions, the same linear response is produced but an angular response is also generated given by (previous reference):

$$\vartheta(S) = - \frac{m\left[\left( l\,\delta c + 2\,c\,\delta l\right) S + l\,\delta k + 2\,k\,\delta l\right]}{\left(J\,S^2 + 2\,c\,l^2\,S + 2\,k\,l^2\right)\left(m\,S^2 + 2\,c\,S + 2\,k\right)} A_F(S) \qquad (34)$$

in which $\delta k \equiv k_2 - k_1$ $\delta c \equiv c_2 - c_1$

and where

$\vartheta(S)$ = Laplace transform of the sensor assembly $\theta$ angular vibration response.

For $A_F(S)$ as an input sinusoid, the amplitudes of the previous acceleration and angular response transfer functions (i.e., the polynomials multiplying $A_F(S)$) are (Ref. 6 (or 9) Sect. 10.6.1):

$$B_A(\Omega) = \sqrt{\frac{\omega_y^4 + 4\,\zeta_y^2\,\omega_y^2\,\Omega^2}{\left(\omega_y^2 - \Omega^2\right)^2 + 4\,\zeta_y^2\,\omega_y^2\,\Omega^2}}$$

$$B_\vartheta(\Omega) = \frac{1}{L}\sqrt{\frac{\omega_\theta^4\left(\varepsilon_k + 4\,\varepsilon_l\right)^2 + 4\,\zeta_\theta^2\,\omega_\theta^2\left(\varepsilon_c + 4\,\varepsilon_l\right)^2\,\Omega^2}{\left[\left(\omega_\theta^2 - \Omega^2\right)^2 + 4\,\zeta_\theta^2\,\omega_\theta^2\,\Omega^2\right]\left[\left(\omega_y^2 - \Omega^2\right)^2 + 4\,\zeta_y^2\,\omega_y^2\,\Omega^2\right]}} \qquad (35)$$

in which

$$\omega_x \equiv \sqrt{\frac{2\,k}{m}} \qquad \zeta_x \equiv \frac{c}{m\,\omega_x} \qquad \omega_\theta \equiv \sqrt{\frac{2\,k\,l^2}{J}} \qquad \zeta_\theta \equiv \frac{c\,l^2}{J\,\omega_\theta}$$
$$\varepsilon_k \equiv \frac{\delta k}{k} \qquad \varepsilon_c \equiv \frac{\delta c}{c} \qquad L \equiv 2\,l \qquad \varepsilon_l \equiv \frac{\delta l}{L} \qquad (36)$$

where

$\Omega$ = $A_F(S)$ sinusoidal input vibration frequency.

$B_A(\Omega)$, $B_\vartheta(\Omega)$ = Magnitudes of the polynomials multiplying $A_F(S)$ in the $A(S)$, $\vartheta(S)$ equations.

Under sinusoidal $A_F(S)$ excitation at frequency $\Omega$, the A(S), $\vartheta$(S) responses would be sinusoidal at frequency $\Omega$ with amplitudes equal to $B_A(\Omega)$, $B_\vartheta(\Omega)$ multiplied by the $A_F(S)$ sinusoid input amplitude, and with generally non-zero phasing relative to the $A_F(S)$ sinusoid (Ref. 6 (or 9) Sect. 10.5.1 also provides the A(S), $\vartheta$(S) phase angle response as a function of $\Omega$).

Although Equations (35) were derived based on the simplified Figure 3 model, they can be applied as universal simplified formulas in which the coefficients and error terms are selected to represent actual sensor-assembly/mount parameters, e.g.,

$\omega_x, \zeta_x$ = Undamped natural frequency and damping ratio for the actual sensor-assembly/mount linear vibration motion dynamic response characteristic.

$\omega_\theta, \zeta_\theta$ = Undamped natural frequency and damping ratio for the actual sensor-assembly/mount rotary vibration motion dynamic response characteristic.

L = Distance between actual sensor assembly mounting points.

$\varepsilon_k, \varepsilon_c$ = Actual sensor assembly mounting structure spring, damping cross-coupling error coefficients.

$\varepsilon_l$ = Distance from the sensor assembly mount center of force to the sensor assembly center of mass, divided by L.

### 3.3 System Response Under Random System Vibration

Section 3.1 described analytical formulas for calculating strapdown INS performance parameters as a function of linear and angular sinusoidal vibrations of the sensor assembly. Section 3.2 described a simplified model of the structural dynamic characteristics for translating a linear sinusoidal vibration input source into resulting linear and angular sinusoidal vibration of the sensor assembly. A typical INS design specification defines the input vibration source as a random mixture of frequency components at frequency dependent amplitudes. The sensor assembly response to random vibration is a composite sum of its response to each frequency component. For the Section 3.1 performance equations, the Section 3.2 simplified sensor assembly dynamical model (interpreted to provide angular response around both axes perpendicular to the linear input vibration), and worst case approximations for phase response of the sensor assembly to vibration excitation, the following can be used to assess system performance under random vibration (Ref. 6 (or 9) Sect. 10.6.1):

$$\mathcal{E}\left(\dot{\Phi}_{Avg}\right) = \int_0^\infty \omega\, B_\vartheta^2(\omega)\, G_{a_{Vib}}(\omega)\, d\omega \quad \text{Coning attitude motion} \qquad (37)$$

$$\mathcal{E}\left(\dot{v}_{Avg}\right) = \int_0^\infty B_\vartheta(\omega)\, B_A(\omega)\, G_{aVib}(\omega)\, d\omega \quad \text{Sculling velocity motion} \qquad (38)$$

$$\mathcal{E}\left(\delta\dot{\Phi}_{Algo}\right) = \mathcal{E}\left(\delta\Delta\dot{\phi}_{ConeAlgo}\right) = \int_0^\infty \omega\, B_\vartheta^2(\omega) \left| \left[1 + \frac{1}{3}\left(1 - \cos \omega T_l\right)\right] \left(\frac{\sin \omega T_l}{\omega T_l}\right) - 1 \right| G_{aVib}(\omega)\, d\omega$$

Attitude/coning algorithm error (39)

$$\left(\delta\dot{v}_{Algo}\right) = \mathcal{E}\left(\delta\Delta\delta\dot{\eta}_{Scull\,Algo}\right) = \int_0^\infty B_\vartheta(\omega)\, B_A(\omega) \left| \left[1 + \frac{1}{3}\left(1 - \cos \omega T_l\right)\right] \left(\frac{\sin \omega T_l}{\omega T_l}\right) - 1 \right| G_{aVib}(\omega)\, d\omega$$

Velocity/sculling algorithm error (40)

$$\mathcal{E}\left(\delta R_{Algo}^2(t)\right) = (t - t_0)^2 \int_0^\infty B_A^2(\omega) \frac{2}{\omega^2} \left\{ E(\omega)^2 + \frac{1}{6}(\omega' T_l)^2 \left[ E(\omega)\, f_1(\omega' T_l) + \frac{1}{12}(\omega' T_l)^2 f_2(\omega' T_l) \right] \right\} f_2\left(\omega'(t - t_0)\right) G_{aVib}(\omega)\, d\omega \qquad (41)$$

Position algorithm folding effect error

$$\omega' \equiv \omega - \frac{2\pi}{T_l}\left(\frac{\omega T_l}{2\pi}\right)_{Intgr} \qquad E(\omega) \equiv \frac{f_1(\omega' T_l)}{2\, f_2(\omega' T_l)} - \frac{\omega'}{\omega}$$

where

$\mathcal{E}(\,)$ = Expected value operator (i.e., average statistical value).

$\omega$ = Input random vibration frequency parameter.

$\omega'$ = Frequency folded version of $\omega$.

$G_{aVib}(\omega)$ = Input linear vibration power spectral density. The integral of $G_{aVib}(\omega)$ from $\omega$ equal zero to plus infinity equals the expected value of the random vibration acceleration input squared.

The $f_1, f_2$ functions, $B_A(\omega)$ and $B_\vartheta(\omega)$ are defined in Equations (32) and (35). Note that $\mathcal{E}\left(\delta R_{Algo}^2(t)\right)$ for the position error is based on the Equation (31) form to avoid singularities when the folded frequency $\omega'$ is zero.

The previous methodology for evaluating particular INS error characteristics under random (and sinusoidal) vibration can be applied to other INS error effects as well. Reference 6 (or 9) Sections 10.6.1-10.6.3 provide several examples in addition to those discussed previously.

## 4. SYSTEM TESTING FOR INERTIAL SENSOR CALIBRATION ERRORS

After an INS (or its sensor assembly) is assembled and sensor compensation software coefficients have been installed (typically based on sensor calibration measurements), it is frequently required that residual sensor error parameters be measured to assess system level performance. For compensatable effects, the results can be used to update the sensor calibration coefficients. This section describes two INS system level tests that are typically conducted in the laboratory for measuring residual bias, scale-factor and misalignment errors: the Strapdown Drift Test and the Strapdown Rotation Test. The Strapdown Drift test is a static test performed on high performance sensor assemblies in which the attitude integration software in the INS computer is configured to constrain the average horizontal transformed specific force acceleration to zero. For a test of several hours duration, the averages of the constraining signals become accurate measures of horizontal angular rate sensor bias error. The Strapdown Rotation Test can be used on sensor assemblies of all accuracy grades. It consists of exposing the INS to a series of rotations, and recording its average transformed specific force acceleration output at static dwell times between rotations. By processing the recorded data, very accurate measurements can be made of the scale factor error and relative misalignment between all inertial sensors in the sensor assembly, the accelerometer bias errors, and misalignment of the sensor assembly relative to the INS mounting fixture. The details of these tests and others are described in Reference 6 (or 9) Chapter 18.

### 4.1 Strapdown Drift Test

The Strapdown Drift test is designed to evaluate angular rate sensor error by processing data generated during extended self-alignment operations. The test is performed on a strapdown analytic platform during an extension of the normal self-alignment initialization mode. The principal measurement of the Strapdown Drift Test is the composite north horizontal angular rate sensor output, determined from the north component of angular rate bias applied to the strapdown analytic platform to render it stationary in tilt around North. Subtracting the known true value of north earth rate from the measurement evaluates the north component of angular rate sensor composite error. East and vertical angular rate sensor errors are ascertained by repeating the test with the previously east and vertical angular rate sensors in the horizontal north orientation.

The self-alignment process utilized in the Strapdown Drift Test creates a locally level rotation rate stabilized analytic "platform" (the N Frame) whose level orientation (relative to the earth) is sustained based on horizontal platform acceleration measurements (i.e., perpendicular to the accelerometer derived local gravity vertical). The test measurement is the biasing rate to the analytically stable platform to maintain it level in the presence of earth's rotation. As configured, the analytic platform remains angularly stable in the presence of B Frame angular rate, hence, angular rate sensor bias determined from stabilized platform measurements becomes insensitive to small physical angular movements of the sensor assembly during the test (caused for example by test-fixture/laboratory micro-motion relative to the earth or rotation of the sensor assembly internal mount (within the INS) due to thermal expansion under thermal exposure testing).

Angular rate sensor bias determined by the previous method is corrupted by angular rate sensor scale factor and misalignment compensation error residuals which are generally negligible in the Strapdown Drift Test environment compared with typical high accuracy bias accuracy requirements. Also contained in the bias measurements are the effects of angular rate sensor random output noise which is reduced to an acceptable level by allowing a long enough extended self-alignment measurement period. If test accuracy requirements permit, a simpler version of the Strapdown Drift Test can be utilized in which the test measurement is the direct integral of the compensated angular rate from each sensor minus its earth rate component input. To reduce earth rate input misalignment error effects using the latter approach, the angular rate sensor can be oriented with its input axis aligned with earth's polar rotation axis. The simpler approach is directly susceptible to angular motion of the sensor assembly relative to the earth during the test measurement.

For situations when the biasing rate to the strapdown analytic platform is not an available INS output, an alternative procedure can be utilized based on INS computed true heading outputs (Ref. 6 (or 9) Sect. 18.2.2). In this case the east angular rate sensor error is determined from the test based on the heading error it generates at the end of an extended self-alignment run. In order to discriminate east angular rate sensor error from North earth rate coupling (under test heading misalignment), the INS heading output is measured for two individual alignment runs. The second alignment run is performed at a heading orientation that is rotated 180 degrees from the first. The difference between the average heading measurements so obtained cancels the North earth rate coupling input, thereby becoming the measurement for east angular rate sensor error determination. North and vertical angular rate sensor errors are ascertained by repeating the test with the previously north and vertical angular rate sensors in the horizontal east orientation.

The following operations are integrated to implement the strapdown analytic platform function during the Strapdown Drift Test extended alignment computational process (Ref. 6 (or 9) Sect. 6.1.2):

$$
\begin{aligned}
\dot{C}_B^N &= C_B^N \left(\underline{\omega}_{IB}^B \times\right) - \left(\underline{\omega}_{IN}^N \times\right) C_B^N \\
\underline{\omega}_{IN}^N &= \underline{\omega}_{IE}^N + \underline{\omega}_{Tilt}^N \\
\underline{\omega}_{Tilt}^N &= K_2 \, \underline{u}_{Up}^N \times \Delta\underline{R}_H^N \\
\underline{\omega}_{IE}^N &= \underline{\omega}_{IE_H}^N + \underline{u}_{Up}^N \, \omega_e \sin l \qquad (42) \\
\dot{\underline{\omega}}_{IE_H}^N &= K_1 \, \underline{u}_{Up}^N \times \Delta\underline{R}_H^N \\
\dot{\underline{v}}_H^N &= \left(C_B^N\right)_H \underline{a}_{SF}^B - K_3 \, \Delta\underline{R}_H^N \\
\Delta\dot{\underline{R}}_H^N &= \underline{v}_H^N - K_4 \, \Delta\underline{R}_H^N
\end{aligned}
$$

where

$\underline{\omega}^{B}_{IB}$, $\underline{a}^{B}_{SF}$ = Angular rate sensor and accelerometer compensated input vectors.

H = Subscript indicating horizontal components (or rows)of the associated vector (or matrix).

$K_i$ = Extended alignment analytical platform level maintenance coefficients.

$\omega_e$ = Earth inertial rotation rate magnitude.

$l$ = Geodetic latitude.

$\underline{u}_{Up}$ = Unit vector upward along the geodetic vertical (i.e., along the N Frame Z axis).

$\underline{v}$, $\Delta\underline{R}$ = Velocity and position displacement during extended alignment.

The North angular rate sensor bias is calculated as an adjunct to the previous operations as (Ref. 6 (or 9) Sect. 18.2.1):

$$\underline{\hat{\phi}}^{\hat{N}}_{H} \equiv \int_{t_{Start}}^{t_{End}} \underline{\omega}^{\hat{N}}_{IN_H}\, dt \qquad \delta\omega_{ARS/Cnst_{North}} \approx \frac{1}{(t_{End} - t_{Start})} \hat{\phi}_H - \omega_e \cos l \qquad (43)$$

where

$t_{Start}$, $t_{End}$ = Time at the start and end of the Strapdown Drift Test measurement period.

$\delta\omega_{ARS/Cnst_{North}}$ = North component of angular rate sensor constant bias residual error.

$\omega_e$ = Earth rotation rate magnitude.

$l$ = Test site latitude.

$\hat{\phi}_H$ = Magnitude of $\underline{\hat{\phi}}_H$.

### 4.2 Strapdown Rotation Test

The basic concept for the Strapdown Rotation Test was originally published by the author in 1977 (Reference 3). Since then, variations of the concept have formed the basis in most strapdown inertial navigation system manufacturing organizations for system level calibration of accelerometer/angular-rate-sensor scale-factors/misalignments and accelerometer biases.

The Strapdown Rotation test consists of a series of rotations of the strapdown sensor assembly using a rotation test fixture for execution. During the test, special software operates on the strapdown angular rate sensor outputs from the sensor assembly to form an analytic angular rate stabilized wander azimuth "platform" (L Frame - See definition to follow) that nominally maintains a constant orientation relative to the earth. The analytic platform is implemented by

processing strapdown attitude-integration/acceleration-transformation algorithms (e.g., Reference 10 Equations (8) - (10), (12) and (24) - (26) including inertial sensor compensation Equations (35) - (43)) with the platform horizontal inertial rotation rate components held constant. Platform horizontal rotation rates are calculated prior to rotation test initiation using special test software that implements strapdown initial alignment algorithms (e.g., Equations (42) using Kalman filter formulated $K_i$ gains). Measurements during the Strapdown Rotation test are taken at stationary positions and computed from the averaged transformed accelerometer outputs plus gravity (i.e., the average computed total acceleration vector):

$$\Delta \underline{v}_m^L \equiv \int_{t_{m-1}}^{t_m} \left( \underline{a}_{SF}^L + \underline{g}^L \right) dt = \Delta \underline{v}_{SF_m}^L - g_{Tst}\, \underline{u}_{Up}^L\, T_m$$

$$\Delta \underline{v}_{SF_m}^L \equiv \int_{t_{m-1}}^{t_m} C_B^L\, \underline{a}_{SF}^B\, dt \qquad (44)$$

$$\underline{a}^L = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \approx \frac{1}{T_m} \Delta \underline{v}_{Avg}^L \qquad \text{Test measurements}$$

where

L Frame = "Attitude Reference" coordinate frame aligned with the N Frame but having Z axis parallel to the downward (rather than upward) vertical and with X, Y axes interchanged (the L Frame X, Y axes are parallel to the N Frame Y, X axes). Reference 6 (or 9) uses the L Frame for "attitude reference" outputs as an intermediate frame between the B and N Frames.

$\underline{g}$ = Plumb-bob gravity vector at the test site (mass attraction "gravitation" plus earth rotation effect centripetal acceleration).

$g_{Tst}$ = Vertical component of $\underline{g}$.

$\Delta \underline{v}_{Avg}^L$ = Output from an averaging process performed on successive $\Delta \underline{v}_{SF_m}^L$'s (See Reference 6 (or 9) Section 18.4.7.3 for process designed to attenuate accelerometer quantization noise).

$\underline{a}$ = Average total acceleration.

a, b, c = Components of $\underline{a}$ in the L Frame.

The fundamental theory behind the Strapdown Rotation test is based on the principle that for a perfectly calibrated sensor assembly, following a perfect initial alignment, the computed L Frame acceleration should be zero at any time the sensor assembly is stationary. Moreover, this should also be the case if the sensor assembly undergoes arbitrary rotations between the time periods that it is set stationary. Therefore, any deviation from zero stationary acceleration can be attributed to imperfections in the sensor assembly (i.e., sensor calibration errors) or in the initial alignment process. Initial alignment process errors create initial L Frame tilt which is removed

from the Strapdown Rotation Test measurements by structuring the horizontal measurements as the difference between average horizontal L Frame acceleration readings taken before and after completing each of the test rotation sequences. As an aside, it is to be noted that in the original Reference 3 paper, the measurement for the rotation test was the average acceleration taken at the end of each rotation sequence, with a self-alignment performed before the start of each rotation sequence. The purpose of the realignment was to eliminate attitude error build-up caused by angular rate sensor error during previous rotation sequences. By taking the measurement as the difference between average accelerations before and after rotation sequence execution (as indicated above), the need for realignment is eliminated. The before/after measurement approach was introduced by Downs in Reference 1 for compatibility with an existing Kalman filter used to extract the acceleration measurements.

The principal advantage for this particular method of error determination derives from the combined use of the angular rate sensors and accelerometers to establish an angular rate stabilized reference for measuring accelerations. This implicitly enables the inertial sensors to measure the attitude of the rotation test fixture settings as the rotations are executed. Consequently, precision rotation test table readout or controls are not required (nor a stable test fixture base), hence, a significant savings can be made in test fixture cost. Inaccuracies in rotation fixture settings manifest themselves as second order errors in sensor error determination, which can be made negligibly small if desired through a repeated test sequence. It has been demonstrated, for example, with precision ring laser gyro strapdown inertial navigation systems, that the test method can measure and calibrate gyro misalignments to better than 1 arc sec accuracy with 0.1 deg rotation fixture orientation inaccuracies. In addition, because the orientation of the sensor assembly is being measured by the sensor assembly itself, it is not necessary that the sensor assembly be rigidly connected to the rotation test fixture. This is an important advantage for high accuracy applications in which the sensor assembly is attached to its chassis and mounting bracket through elastomeric isolators of marginal attitude stability.

While most of the sensor calibration errors evaluated by the Strapdown Rotation test can be measured on an individual sensor basis, the rotation test is the only direct method for measuring relative misalignments between the sensor input axes. It should also be noted that determination of sensor-assembly-to-mount misalignment is not an intrinsic part of the Strapdown Rotation Test, however, because the data taken during the test allows for this determination, it is easily included as part of test data processing (Ref. 6 (or 9) Sect. 18.4.5).

Reference 6 (or 9) Section 18.4 (and subsections) provides a detailed description of the Strapdown Rotation Test, its analytical theory, processing routines, and structure based on two sets of rotation sequences (a 16 rotation sequence set and a 21 rotation sequence set). The rotation sequences for the 16 set are summarized in Table 1.

**Table 1 - 16 Set Rotation Test Sequences**

| SEQUENCE NUMBER | ROTATION SEQUENCE (Degrees, B Frame Axis) | STARTING ATTITUDE (+Z Down, Axis Indicated Along Outer Rotation Fixture Axis) |
|---|---|---|
| 1 | +360 Y | +Y |
| 2 | +360 X | +X |
| 3 | +90 Y, +360 Z, -90 Y | +Y |
| 4 | +180 Y, +90 Z, +180 X, -90 Z | +Y |
| 5 | +180 X, +90 Z, +180 Y, -90 Z | +X |
| 6 | +90 Y, +90 Z, -90 X, -90 Z | +Y |
| 7 | +90 Y | +Y |
| 8 | -90 Y | +Y |
| 9 | +90 Y, +90 Z | +Y |
| 10 | +90 Y, -90 Z | +Y |
| 11 | -90 Y, -90 Z | +Y |
| 12 | +90 X, +90 Z | +X |
| 13 | +90 X, -90 Z | +X |
| 14 | +180 Z | +Y |
| 15 | +180 Y | +Y |
| 16 | +180 X | +X |

Based on the Table 1 rotation sequences, Reference 6 (or 9) Section 18.4.3 develops the relationship between the test measurements and the sensor errors excited by the test; e.g., for Table 1 rotation sequences 1 and 9:

$$
\begin{aligned}
\Delta a_1 &= -2\,\pi\, g\, \kappa_{yy} \\
\Delta b_1 &= 0 \\
c_1^1 = c_1^2 &= -g\left(\lambda_{zz} - \lambda_{zzz}\right) + \alpha_z
\end{aligned}
\qquad
\begin{aligned}
\Delta a_9 &= -g\left(\frac{1}{2}\upsilon_{zx} + \frac{1}{2}\upsilon_{yz} + \mu_{zy} - \mu_{xz} + \frac{\pi}{2}\kappa_{yy}\right) \\
&\quad - \alpha_x + \alpha_z \\
\Delta b_9 &= -g\left(\frac{1}{2}\upsilon_{xy} + \frac{1}{2}\upsilon_{yz} + \mu_{yz} - \mu_{xy} + \frac{\pi}{2}\kappa_{zz}\right) \\
&\quad + \alpha_x - \alpha_y \\
c_9^1 &= -g\left(\lambda_{zz} - \lambda_{zzz}\right) + \alpha_z \\
c_9^2 &= -g\left(\lambda_{yy} - \lambda_{yyy}\right) + \alpha_y
\end{aligned}
\qquad (45)
$$

where

$\Delta a_i$, $\Delta b_i$ = Difference between a, b horizontal acceleration measurements taken at the start and end of rotation sequence i.

$c_i^1$, $c_i^2$ = Vertical acceleration measurements taken immediately before (superscript 1) and after (superscript 2) rotation sequence i.

$\alpha_i$ = i axis accelerometer bias calibration error.

$\lambda_{ii}$ = i axis accelerometer symmetrical scale factor calibration error.

$\lambda_{iii}$ = i axis accelerometer scale factor asymmetry calibration error.

$\kappa_{ii}$ = i axis angular rate sensor scale factor calibration error.

$\upsilon_{ij}$ = Orthogonality compensation error between the i and j angular rate sensor input axes, defined as $\pi/2$ radians minus the angle between the compensated i and j sensor input axes.

$\mu_{ij}$ = i axis accelerometer misalignment calibration error, coupling specific force from the j axis of the mean angular rate sensor axes into the i axis accelerometer input axis.

The mean angular rate sensor (MARS) axis frame in the previous $\mu_{ij}$ definition refers to a B Frame defined as the orthogonal triad that best fits symmetrically within the actual compensated angular rate sensor input axes. The "best fit" condition is specified as the condition (measured around angular rate sensor axis k) for which the angle between angular rate sensor input axis i and MARS axis i equals the angle between angular rate sensor input axis j and MARS axis j (Ref. 6 (or 9) Sect. 18.4.3). As such, the overall angular misalignment of the actual angular rate sensor triad is defined to be zero relative to the MARS frame, and individual angular rate sensor misalignments affecting the Strapdown Rotation Test measurements are only due to orthogonality errors between the angular rate sensor axes.

Once the $\Delta a_i$, $\Delta b_i$, $c_i^1$, $c_i^2$ measurements are obtained, the individual sensor residual errors can be calculated deterministically as summarized in Figure 4 (Ref. 6 (or 9) Sect. 18.4.4).

ANGULAR RATE SENSOR CALIBRATION ERRORS

Scale Factor Errors

$$\kappa_{xx} = -\frac{1}{2\pi g}\Delta a_2$$

$$\kappa_{yy} = -\frac{1}{2\pi g}\Delta a_1$$

$$\kappa_{zz} = \frac{1}{2\pi g}\Delta b_3$$

Orthogonality Errors

$$\upsilon_{xy} = \frac{1}{g}\left(\Delta b_6 - \frac{1}{2}\Delta b_4 - \frac{1}{4}\Delta b_3\right)$$

$$\upsilon_{yz} = \frac{1}{4g}\left(\Delta b_5 + \Delta b_4\right)$$

$$\upsilon_{zx} = \frac{1}{4g}\left(\Delta b_5 - \Delta b_4\right)$$

ACCELEROMETER CALIBRATION ERRORS

Bias Errors

$$\alpha_x = \frac{1}{4}\Delta a_1 - \frac{1}{2}\Delta a_{15}$$

$$\alpha_y = \frac{1}{2}\Delta a_{16} - \frac{1}{4}\Delta a_2$$

$$\alpha_z = \frac{1}{2}\left(\Delta a_7 - \Delta a_8\right) - \frac{1}{4}\Delta a_1$$

Scale Factor Errors

$$\lambda_{xx} = -\frac{1}{2g}\left(c_{12}^2 + c_{13}^2\right)$$

$$\lambda_{yy} = -\frac{1}{2g}\left(c_9^2 + c_{10}^2\right)$$

$$\lambda_{zz} = -\frac{1}{2g}\left(c_{14}^2 + c_{15}^2\right)$$

Scale Factor Asymmetry

$$\lambda_{xxx} = \frac{1}{2g}\left(c_{12}^2 - c_{13}^2 + \Delta a_{15} - \frac{1}{2}\Delta a_1\right)$$

$$\lambda_{yyy} = \frac{1}{2g}\left(c_9^2 - c_{10}^2 - \Delta a_{16} + \frac{1}{2}\Delta a_2\right)$$

$$\lambda_{zzz} = \frac{1}{2g}\left(c_{14}^2 - c_{15}^2 - \Delta a_7 + \Delta a_8 + \frac{1}{2}\Delta a_1\right)$$

Misalignment Relative To Mean Angular Rate Sensor Axes

$$\mu_{xy} = \frac{1}{2g}\left(\Delta b_{11} - \Delta b_{10} + \Delta b_6 - \frac{3}{4}\Delta b_3 - \frac{1}{2}\Delta b_4\right)$$

$$\mu_{yx} = \frac{1}{2g}\left(\Delta b_7 - \Delta b_8 - \Delta b_6 + \frac{1}{2}\Delta b_4 + \frac{1}{4}\Delta b_3\right)$$

$$\mu_{yz} = \frac{1}{2g}\left(\Delta b_{14} + \Delta a_{16} - \frac{1}{2}\Delta a_2 + \frac{1}{4}\Delta b_4 + \frac{1}{4}\Delta b_5\right)$$

$$\mu_{zy} = \frac{1}{2g}\left(\Delta a_{10} - \Delta a_9 - \frac{1}{4}\Delta b_4 - \frac{1}{4}\Delta b_5\right)$$

$$\mu_{zx} = \frac{1}{2g}\left(\Delta a_{13} - \Delta a_{12} - \frac{1}{4}\Delta b_5 + \frac{1}{4}\Delta b_4\right)$$

$$\mu_{xz} = \frac{1}{2g}\left(\Delta a_{14} - \Delta a_{15} + \frac{1}{2}\Delta a_1 + \frac{1}{4}\Delta b_5 - \frac{1}{4}\Delta b_4\right)$$

**Figure 4 - Sensor Errors In Terms Of Measurements For The 16 Rotation Sequence Test**

The Figure 4 results can then be used to update the INS sensor calibration coefficients (Ref. 6 (or 9) Sect. 18.4.6). If the B Frame is chosen to be the MARS Frame as described previously, the $\mu_{ij}$ accelerometer misalignments calculated from Figure 4 would be used directly to update the accelerometer misalignment calibration coefficients relative to the B Frame. For the angular rate sensors, selecting the B Frame as the MARS Frame equates to the following for individual angular rate sensor misalignments relative to the B Frame as:

$$\kappa_{xy} = \kappa_{yx} = \frac{1}{2}\upsilon_{xy} \qquad \kappa_{yz} = \kappa_{zy} = \frac{1}{2}\upsilon_{yz} \qquad \kappa_{zx} = \kappa_{xz} = \frac{1}{2}\upsilon_{zx} \tag{46}$$

where

$\kappa_{ij}$ = Angular rate sensor misalignment calibration error coupling B Frame j axis angular rate into the i angular rate sensor input axis.

## 5. SYSTEM PERFORMANCE ANALYSIS

To assess the accuracy of inertial navigation systems, error analysis techniques are traditionally employed in which error equations are used to describe the propagation of system navigation error parameters in response to system error sources. The error equations also form the basis for performance improvement techniques in which the inertial system errors are estimated and controlled in real time based on navigation measurements taken from other navigation devices (e.g., GPS satellite range measurements). Such "aided" inertial navigation systems are structured using a Kalman filter in which system error estimates are based on a running statistical determination of the expected instantaneous errors (e.g., typically in the form of a "covariance matrix"). The covariance matrix computational structure used in the Kalman filter is also applied in "covariance analysis" simulators to statistically analyze both aided and unaided ("free inertial") system performance. Validation of the Kalman filter software is an important element in the aided inertial navigation system software design process.

### 5.1 Free Inertial Performance Analysis

The accuracy of all inertial navigation systems is fundamentally limited by instabilities in the inertial component error characteristics following calibration. Resulting residual inertial sensor errors produce INS navigation errors that are unacceptable in many applications. To predict Strapdown INS performance, linear time rate differential error propagation equations can be analyzed depicting the growth in INS computed attitude, velocity, position error as a function of residual inertial sensor and gravity modeling error (e.g., Ref. 10 Eqs. (51)). Modern formulations of such error propagation equations cast them in a standard error state dynamic equation format as follows (Ref. 2 Sect. 3.1 and Ref. 6 (or 9) Sect. 15.1):

$$\dot{\underline{x}}(t) = A(t)\,\underline{x}(t) + G_P(t)\,\underline{n}_P(t) \tag{47}$$

where

$\underline{x}(t)$ = Error state vector treated analytically as a column matrix.

$A(t)$ = Error state dynamic matrix.

$\underline{n}_P(t)$ = Vector of independent white "process" spectral noise density sources driving $\underline{x}(t)$ (treated analytically as a column matrix).

$G_P(t)$ = Process noise dynamic coupling matrix that couples individual $\underline{n}_P(t)$ components into $\dot{\underline{x}}(t)$ .

In general, $A(t)$ and $G_P(t)$ are time varying functions of the angular rate, acceleration, attitude, velocity and position parameters within the INS computer. To evaluate the solution to Equation (47) at discrete time instants, the following equivalent integrated form is utilized (Ref. 2 Sect. 3.4 and Ref. 6 (or 9) Sect. 15.1.1):

$$\underline{x}_n = \Phi_n \underline{x}_{n-1} + \underline{w}_n \qquad (48)$$

in which

$$\Phi(t, t_{n-1}) = I + \int_{t_{n-1}}^{t} A(\tau)\, \Phi(\tau, t_{n-1})\, d\tau \qquad \Phi_n \equiv \Phi(t_n, t_{n-1}) \qquad (49)$$

$$\underline{w}_n = \int_{t_{n-1}}^{t_n} \Phi(t_n, \tau)\, G_P(\tau)\, \underline{n}_P(\tau)\, d\tau \qquad (50)$$

where

$n$ = Performance evaluation cycle time index.

$\underline{x}_n$ = Error state vector evaluated at cycle time n.

$\Phi_n$ = Error state transition matrix that propagates the error state vector from the $n\text{-}1^{th}$ to the $n^{th}$ time instant.

$\underline{w}_n$ = Change in $\underline{x}_n$ due to process noise input from the $n\text{-}1^{th}$ to the $n^{th}$ time instant.

For a strapdown INS, the elements of the $\underline{x}$ error state vector would include INS attitude, velocity, position error parameters, inertial sensor error parameters (e.g., bias, scale factor, misalignment) and gravity modeling error. Elements of the $\underline{n}_P$ process noise vector would include inertial sensor random output noise, noise source input to randomly varying inertial sensor error states, and noise source inputs to randomly varying gravity error modeling error states. Equations (51) of Reference 10 are an example of strapdown INS error propagation equations that are in the Equation (47) form. The sensor error terms in these equations are typically modeled as random constants (with random walk input white noise), first order Markov processes, or the sum of both (Ref. 6 (or 9) Sect. 12.5.6). Reference 6 (or 9) Section 16.2.3.3 provides an example of how the gravity error term in these equations can be modeled.

## 5.2 Kalman Filters For INS Aiding

To overcome the performance deficiencies in a free inertial navigation system, "inertial aiding" is commonly utilized in which the INS navigation parameters (and in some cases, the sensor calibration coefficients) are updated based on inputs from an alternate source of navigation information available in the user vehicle. The modern method for applying the inertial aiding measurement to the INS data is through a Kalman filter, a set of software that is typically resident in the INS computer. The Kalman filter is designed based on the Equation (48) $\underline{x}$ error state vector propagation model, to generate estimates for $\underline{x}$ and provide updates to the INS computer parameters to control $\underline{x}$ (ideally to zero). For an aided INS, the $\underline{x}$ error state vector would also include error terms associated with the aiding device. The basic structure of a real-time Kalman filter based on "delayed control resets" (to allow for finite computation time delay - Ref. 6 (or 9) Sect. 15.1.2) is:

$$\xi_{INS_n}(+_c) = \xi_{INS_n}(-) + g_{INS}\left(\xi_{INS_n}(-), \underline{u}_{c_n}\right)$$
$$\xi_{Aid_n}(+_c) = \xi_{Aid_n}(-) + g_{Aid}\left(\xi_{INS_n}(-), \underline{u}_{c_n}\right) \qquad (51)$$

$$\underline{Z}_{Obs_n} = f\left(\xi_{INS_n}(+_c), \xi_{Aid_n}(+_c)\right) \qquad (52)$$

$$\tilde{\underline{x}}_n(-) = \Phi_n \tilde{\underline{x}}_{n-1}(+_e) \qquad (53)$$

$$\tilde{\underline{x}}_n(+_c) = \tilde{\underline{x}}_n(-) + \underline{u}_{c_n} \qquad (54)$$

$$\tilde{\underline{z}}_n = H_n \tilde{\underline{x}}_n(+_c) \qquad (55)$$

$$\tilde{\underline{x}}_n(+_e) = \tilde{\underline{x}}_n(+_c) + K_n\left(\underline{Z}_{Obs_n} - \tilde{\underline{z}}_n\right) \qquad (56)$$

$$\underline{u}_{c_{n+1}} = \text{function of } \tilde{\underline{x}}_n(+_e) \qquad (57)$$

$$\tilde{\underline{x}}_0 = 0 \qquad \text{Initial Conditions} \qquad (58)$$

where

$\xi_{INS}$ = INS navigation parameters.

$\xi_{Aid}$ = Aiding device navigation parameters.

$g_{INS}(\ ), g_{Aid}(\ )$ = Non-linear functional operators used to apply $\underline{u}_{c_n}$ to the $\xi_{INS}$, $\xi_{Aid}$ navigation parameters at time $t_n$ such that the error in these parameters is controlled (typically to zero).

$f(\ )$ = Functional operator that compares designated equivalent elements of $\xi_{INS}$ and $\xi_{Aid}$. The $f(\ )$ operator is designed so that for an error free INS, an error free aiding device, and a perfect (error free) $f(\ )$ software implementation, $f(\ )$ will be zero.

$\underline{Z}_{Obs}$ = Observation vector formed from the comparison between comparable INS and aiding device navigation parameters.

$\underline{u}_{c_{n+1}}$ = Control vector derived from the Kalman filter estimate of the time $t_n$ value of $\underline{x}$ and applied at time $t_{n+1}$ to constrain the actual value of $\underline{x}$.

$\tilde{\ }$ = Value for parameter estimated (or predicted) by the Kalman filter.

$(+_e)$ = Designation for parameter value at its designated time stamp ($t_n$ in this case) immediately after ("a posteriori") the application of estimation resets (e subscript) at the same designated time.

$(+_c)$ = Designation for parameter value at its designated time stamp ($t_n$ in this case) immediately after ("a posteriori") the application of control resets (c subscript) at the same designated time.

(-) = Designation for parameter value at its designated time stamp ($t_n$ in this case) immediately prior to ("a priori") the application of any resets (estimation or control) at the same designated time.

$K_n$ = Errors state estimation gain matrix.

n = Kalman filter software cycle time index.

$(\ )_n$ = $(\ )$ at the $n^{th}$ Kalman filter cycle time.

$\tilde{\underline{z}}$ = Estimated "measurement vector" analytically represented as a column matrix. The $\tilde{\underline{z}}$ equation implemented in the Kalman filter represents a linearized version of the $\underline{Z}_{Obs}$ observation equation based on the expected (projected) value of the error state vector $\underline{x}$ when $\underline{Z}_{Obs}$ is measured.

H = The "measurement matrix". Generally a time varying function of the navigation parameters calculated in the INS computer. See further description in the paragraph following Equation (59) parameter definitions.

The previous Latin notation "a priori" and "a posteriori" has been adopted in Kalman filter terminology to add an element of "mysterioso". Identification of individual $(+_e)$ and $(+_c)$ "a posteriori" updates provides flexibility to allow for different Kalman filter estimation/control time points (e.g., for timing and synchronization of observation/measurement/control operations - Ref. 6 (or 9) Sect. 15.1.2.4).

The estimation process described by Equation (56) is general and becomes a Kalman filter operation when the gain matrix $K_n$ matrix is computed based on "optimally" estimating the error state vector as follows (Ref. 2 Sect. 4.2 and Ref. 6 (or 9) Sect. 15.1.2.1):

$$K_n = P_n(-)\, H_n^T \left( H_n\, P_n(-)\, H_n^T + G_{M_n}\, R_n\, G_{M_n}^T \right)^{-1} \qquad (59)$$

in which

$$P \equiv \mathcal{E}\left( \Delta\underline{x}\, \Delta\underline{x}^T \right) \qquad \Delta\underline{x} \equiv \tilde{\underline{x}} - \underline{x} \qquad R_n \equiv \mathcal{E}\left( \underline{n}_{M_n}\, \underline{n}_{M_n}^T \right)$$

where

$\Delta\underline{x}$ = Error state vector estimation uncertainty.

$P$ = Error state vector uncertainty covariance matrix.

$\underline{n}_M$ = Vector of independent white measurement noise sources (represented analytically as a column matrix). The $\underline{n}_M$ vector represents noise type error effects that may be introduced in the process of making the $\underline{Z}_{Obs}$ observation.

$G_M$ = Measurement noise dynamic coupling matrix that couples $\underline{n}_M$ into the $\underline{Z}_{Obs}$ observation. Generally a time varying function of the navigation parameters calculated in the INS computer.

From an analytical standpoint, $G_M$ and $\underline{n}_M$ in Equation (59) (and H in Equation (55)) are defined as part of $\underline{z}$, the linearized analytical form of the $\underline{Z}_{Obs}$ observation, which is denoted as the "measurement equation": $\underline{z}_n = H_n\, \underline{x}_n + G_{M_n}\, \underline{n}_{M_n}$ (Ref. 6 (or 9) Sect. 15.1 and Ref. 2 Sect. 3.5). The covariance matrix P in Equation (59) is calculated by an integration operation based on the statistical uncertainty in the Equation (56) estimation process (using the previous $\underline{z}_n$ approximation for $\underline{Z}_{Obs_n}$) and the Equation (53) approximation for the actual Equation (48) error state vector propagation between estimation cycles (Ref. 2 Sect. 4.2 and Ref. 6 (or 9) Sects. 15.1.2.1 and 15.1.2.1.1):

$$P_n(-) = \Phi_n\, P_{n-1}(+_e)\, \Phi_n^T + Q_n \qquad (60)$$

$$P_n(+_e) = \left( I - K_n\, H_n \right) P_n(-) \left( I - K_n\, H_n \right)^T + K_n\, G_{M_n}\, R_n\, G_{M_n}^T\, K_n^T \qquad (61)$$

in which

$$Q_n \equiv \mathcal{E}\left( \underline{w}_n\, \underline{w}_n^T \right)$$

### 5.2.1 Covariance Simulation Analysis

The computational structure used in computing the Kalman filter covariance matrix (Eqs. (59) - (61)) can also be used in performance analysis time domain simulation programs for statistically estimating aided INS accuracy (or unaided performance by setting the $K_n$ gain matrix to zero). Such covariance simulation programs (Ref. 6 (or 9) Chpt. 16) are commonly used to provide numerical time histories depicting the accuracy of a given system configuration

in terms of the covariance of its associated linearized error state vector. For a Kalman filter aided system, the covariance simulation is also utilized as a basic design tool during the synthesis and test of the "suboptimal" Kalman filter configuration used in the actual system. The suboptimal Kalman filter configuration is typically based on a simplified error state dynamic/measurement model (compared to the "real world" error state dynamics/measurements) with numerical values for its defining matrix elements that may differ from real world values. The covariance simulation is used to evaluate the performance of the suboptimal filter operating in a real world environment, and to provide the design engineer with useful sensitivities for identifying sources of undesirable performance characteristics during the design process.

### 5.3 Kalman Filter Validation

Although a Kalman filter is generally a complex software package, its validation process can be fairly straight-forward because of its fundamental underlying structure. The Kalman filter elements are well defined analytically and can be validated individually based on their intrinsic properties. Once the elements are validated, the proper operation of the filter is assured through its theoretical structure.

As an example, Reference 6 (or 9) Section 15.1.4 discusses the following operations that can be performed using specialized test simulators for validating the Equations (51) - (58) and (59) - (61) Kalman filter algorithms:

- The state transition matrix $\Phi_n$, estimated measurement $\underline{\tilde{z}}_n$, and observation $\underline{Z}_{Obs_n}$ algorithms can be validated by operating Equations (51) - (58) "open loop" (i.e., setting the Kalman gain $K_n$ and control vector $\underline{u}_c$ to zero) using simulators for $\xi_{INS_n}$ and $\xi_{Aid_n}$. The $\xi_{INS_n}$ simulator would consist of the strapdown inertial navigation algorithms upon which $\Phi_n$ is based. The $\xi_{Aid_n}$ simulator would be built onto a previously validated trajectory generator; the trajectory generator would also provide the strapdown inertial sensor inputs to $\xi_{INS_n}$. The Kalman filter error state vector $\underline{\tilde{x}}_n$ components would be initialized to some arbitrary non-zero value; the same error values would be inserted into the $\xi_{INS_n}$, $\xi_{Aid_n}$ parameters. Under these conditions, the Kalman filter estimated measurement $\underline{\tilde{z}}_n$ calculated with (55) should track the observation vector $\underline{Z}_{Obs_n}$ computed with (52), resulting in a zero value for the measurement residual $\underline{Z}_{Obs_n} - \underline{\tilde{z}}_n$ (within the fundamental linearization error in $\underline{\tilde{z}}_n$). A zero measurement residual validates the $\Phi_n$, $\underline{\tilde{z}}_n$ and $\underline{Z}_{Obs_n}$ algorithms and associated timing structure in the simulation implementation.

- The covariance propagation algorithm (with process noise set to zero) can be validated as part of the previous process by initially setting the covariance matrix equal to the arbitrarily defined $\underline{\tilde{x}}_n$ error state vector times its transpose. The covariance matrix would then be propagated without resets using the Equation (60) algorithm or a Reference 6 (or

9) Section 15.1.2.1.1.3 equivalent (several propagation cycles between estimation cycles). The propagated covariance matrix should then equal the propagated error state vector times its transpose.

- The algorithms for calculating the Kalman gain matrix $K_n$ and resetting the covariance matrix can be validated by comparing the covariance reset algorithm output with the output from an equivalent alternative algorithm based on the analytical form of $K_n$ (e.g., the Equation (61) "Joseph's" form compared with the Reference 6 (or 9) Equation (15.1.2.1.1-4) optimal form). The results should be identical.

- The basic estimation capability of the Kalman filter can be validated by disabling the control vector (setting $\underline{u}_c$ to zero) and allowing the Kalman filter to estimate $\underline{\tilde{x}}_n$ in the presence of selected values for the error state components initially imbedded in $\xi_{INS_n}$ and $\xi_{Aid_n}$. For this test, the process and measurement noise matrices in the Kalman filter covariance propagation/reset routines would be set to zero to heighten sensitivity (and better account for the error condition being simulated).

- Kalman filter estimation capability in the presence of process and measurement noise can be validated by repeating the previous test, but with random noise (from a software noise generator at the Kalman filter specified white noise source amplitudes) applied appropriately to the $\xi_{INS_n}$, $\xi_{Aid_n}$ models (for process noise) and to the $\underline{Z}_{Obs_n}$ routine (for measurement noise). The Kalman filter process and measurement noise matrices would also be active for this test. In parallel, a "truth model" error state vector history would be generated using the same noise and initial conditions applied to a simulated version of error state dynamic Equation (47). The uncertainty in the Kalman filter estimated error state vector is evaluated by comparing the filter error state vector estimate with the "truth model" error state vector. Repeated runs with different random noise generator initial "seeds" provides an ensemble history of the error state uncertainty. The ensemble average of the uncertainty times its transpose (at common time points) should match the corresponding filter covariance matrix history.

- The control vector $\underline{u}_c$ interface in control reset Equations (51) and (54) can be validated by assigning an arbitrary value to $\underline{u}_c$ and applying it to the previous equations. If the control reset equations and the measurement/observation algorithms are consistent, the measurement residual $\underline{Z}_{Obs_n} - \underline{\tilde{z}}_n$ should be unaffected by the control reset application.

A previously validated trajectory generator is an important supporting software element in the Kalman filter validation process to provide truth model navigation parameter data over a user shaped trajectory profile. A trajectory generator is also required in covariance analysis programs to provide navigational parameters for computing the error-state-transition, measurement and noise matrices. Reference 6 (or 9) Chapter 17 describes a trajectory generator based on exact strapdown inertial navigation integration algorithms that can be validated using the steps outlined in Section 2 (and subsections) of this paper.

## 6. SYSTEM INTEGRATION TESTING

System performance testing is conducted to verify that the system meets accuracy requirements under anticipated user environments (e.g., temperature, vibration, altitude, etc.). Prior to performance testing, system integration testing must be conducted to verify that functional operations are performed properly and accurately by all hardware, software, and interface elements. Based on direct experience, it is the author's firm contention that all software operations should be (and can be) completely validated prior to hardware/software integration of a strapdown INS. Otherwise, problems that will inevitably be encountered during final system integration (e.g., software errors due to programming flaws or algorithm error) may never be completely resolved (e.g., hardware designers may fault the software, software designers may fault the hardware - particularly the inertial components, thus discouraging meaningful problem resolution). For an aided strapdown INS, the software validation procedures discussed in Sections 2 and 5.3 can be utilized.

Hardware/software integration begins with software/system-computer integration. The purpose should be to verify identical performance in the system computer (within minuscule round-off error) as achieved in the computer used for software validation. Toward this end, the same simulators/truth-models used for the software validation process would be installed with the system software being integrated as the computer/software integration test driver/evaluator. The driver/evaluator should be designed/validated (as part of the software validation process) to fully exercise/verify all system software under simulated system inputs. In this regard, the driver/evaluator should be considered to be an integral part of the validated system software. For today's computer technology with associated high speed floating point architecture, long word-length, large memory capabilities and abundant software compiler/translator tools, computer/software integration should be a fairly straightforward task.

Hardware integration precedes hardware/software integration based on traditional methods in which functional elements are first individually tested, then interfaced/tested in functional groups until a fully integrated hardware assembly is verified. A critical part of hardware integration is the individual testing (by applied stimulus) of all functional element input/output interfaces to verify that proper signals are being transmitted to assigned locations with proper phasing. Analog signal inputs to the system computer must be individually tested to assure proper error free analog-to-digital conversion. Digital computer interfaces should be individually checked to assure immunity to system self-generated electrical noise and externally applied electro-magnetic interference (EMI). For a strapdown INS, common internal computer interfaces to be tested are for inertial sensor inputs, for individual temperature probe inputs (used for temperature sensitive sensor compensation software), and for special computer input/output signals used to control individual internal sensor operations (e.g., path length control resets for ring laser gyros). Successful interface testing requires pre-planning in the hardware design process for the ability to stimulate all interfaces to be tested. For an aided INS, interfaces with the aiding device must also be verified (e.g., GPS data).

A powerful technique for demonstrating satisfactory completion of the strapdown INS hardware/software integration process is to execute a system level laboratory calibration procedure (e.g., using the Section 4.1 and 4.2 Strapdown Rotation Test and Strapdown Drift

Test). The system should perform accurately after re-calibration based on the test results. For a GPS aided INS, a successful GPS data interface can be demonstrated by a correct GPS data based position solution generated independently within the integrated system computer (compared with the same solution generated externally using an independent GPS receiver system).

For recent GPS aided INS micro-electronic "deeply integrated" architectures designed around MEMS (micro-machined electro-mechanical systems) inertial sensors, application of the previous integration test techniques poses new challenges.

## 7. CONCLUDING REMARKS

Strapdown inertial navigation computation algorithms can be accurately validated using simple closed-form exact solution truth models for reference. Algorithm validation can be greatly facilitated by structuring the algorithms based on exact integral solutions between update cycles of the continuous form navigation parameter time rate differential equations. This permits the algorithms to be validated using simple generic application independent truth models designed to exercise all algorithm elements. The truth models generally do not have to simulate realistic trajectory profiles.

Vibration induced inertial system error effects are easily analyzed using simplified analytical INS structural dynamic models. Simplified simulators based on these models can quickly generate numerical system performance measurements (e.g., coning/sculling motion, coning/sculling algorithm error, position integration algorithm folding effect error, vibration induced sensor error) as a function of system vibration power spectrum input, sensor assembly mounting dynamics/imbalance, and algorithm update frequency.

Several methods are available for INS system level performance analysis in the test laboratory to evaluate residual sensor errors remaining after system calibration. The Strapdown Drift Test and Strapdown Rotation Test provide simple methods for accurately measuring residual strapdown inertial sensor calibration errors without requiring elaborate precision test fixturing.

Kalman filters for strapdown INS aiding should be validated based on their natural internal structure using a simulated version of the INS being aided (interfaced to the Kalman filter) and a simulated aiding device. The software in the simulated INS should be validated prior to Kalman filter testing. Inputs to the INS and aiding device simulators would be provided by a previously validated trajectory generator. A trajectory generator is also required for covariance simulation analysis performance assessment of aided and unaided inertial navigation systems. Trajectory generator validation can be performed using the same methods used to validate the INS software.

System software should be thoroughly validated prior to system integration testing.

**REFERENCES**


1. Downs, H. B., "A Lab Test To Find The Major Error Sources In A Laser Strapdown Inertial Navigator", 38th Annual Meeting of the ION, Colorado Springs, CO, June 15-17, 1982.

2. Gelb, A., *Applied Optimal Estimation*, The MIT Press, Cambridge Mass., London, England, 1978.

3. Savage, P. G., "Calibration Procedures For Laser Gyro Strapdown Inertial Navigation Systems", 9th Annual Electro-Optics / Laser Conference and Exhibition, Anaheim, California, October 25-27, 1977.

4. 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.

5. 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.

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

7. Savage, P. G., "Strapdown System Performance Analysis", *Advances In Navigation Sensors and Integration Technology*, NATO RTO Lecture Series No. 232, October 2003, Section 4.

8. 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.

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

10. Savage, P. G., "Computational Elements For Strapdown Systems", *Low Cost Navigation Sensors and Integration Technology*, NATO RTO-EN-SET-116(2008), Section 9, published in 2009, also reformatted for Web Open Access as SAI-WBN-14010, May 31, 2015 at wwww.strapdownassociates.com.