SKEWED SENSOR FAILURE DETECTION USING PARALLEL NAVIGATION SOLUTIONS

Paul G. Savage
Strapdown Associates, Inc.

SAI WBN-14017
www.strapdownassociates.com
June 16, 2016

ABSTRACT

This article describes a skewed sensor redundancy approach based on navigation parity vectors, each formed by comparing dual inertial navigation solutions, each generated using inputs from a triad of 3 skewed inertial sensor sets (gyros and accelerometers). Each set of dual triads is defined from a 4 sensor skewed tetrad within an overall skew redundant sensor array. All combinations of tetrads are selected within the skewed array to generate multiple navigation parity vectors for navigation-solution failure detection. Navigation outputs are formed from navigation solution inputs to parity vectors that pass validity tests (i.e., having all components less than prescribed navigation error limits). For an n skewed sensor array there are n! / (4! (n - 4)!) tetrads. Modern day computer technology makes the associated throughput/memory requirement easily achievable. The skewed sensor parallel navigation solution concept is illustrated using a skewed 6-axis (hexad) sensor array. It is shown by example how 12 parallel navigation solutions can be used to form navigation parity vectors for all 15 possible tetrads in the hexad. Navigation solution failure detection/isolation logic is described for the hexad, showing the ability to detect/isolate the first 2 navigation solution failures, provide valid navigation outputs with 2 navigation solution failures, and to detect the occurrence of a third failure.

INTRODUCTION

A commonly used method for increasing reliability in electronic systems is through the use of redundant elements. For an Inertial Navigation System (INS), a classical approach to redundancy has been by direct system duplication. With this approach, navigation outputs from two INSs are compared against accuracy limits. Falling within test limits indicates that the two INSs are operating properly; falling outside of test limits indicates that one of the two has experienced a failure ("soft" or "hard", depending on the magnitude of the miscompare). Identification of the failure to a particular INS is accomplished by comparing both INS outputs in the failed set to a third INS. The failed INS is then identified as the one that fails the comparison test with the third INS (given that the other INS in the failed set passes a comparison test with the third). Based on this logic, for n INSs, the occurrence of failures in n-1 INSs can be detected, and failures in n-2 INSs can be isolated to the failed unit. Thus, to isolate up to two failed INSs, four INSs would be required (quad redundancy).

Original redundant INSs were implemented using gimbaled platforms to isolate inertial sensors from vehicle rotation. With the advent of strapdown inertial navigation technology, a computational process in the INS computer replaced the mechanical platform, allowing the equivalent of “platform” mounted sensor signals to be created within the INS computer using vehicle ("body") mounted ("strapdown") inertial sensor signals for input. The strapdown sensor signals provide additional potential benefits not readily available in traditional gimbaled systems: 1) Use of the body mounted sensor signals for autopilot functions traditionally provided by a separate group of flight control gyros/accelerometers, and 2) A new form of INS redundancy based on mounting the inertial sensors in a non-orthogonal configuration.

In the past, 3-axis (triad) body mounted inertial sensor clusters were configured with input axes orthogonal. Each level of redundancy was achieved by duplicating sensor triads (6 sensors for dual redundancy, 9 sensors for triple redundancy, 12 sensors for quad redundancy). In effect, this was the approach taken with gimbaled systems. It can be analytically demonstrated, however, that orthogonal sensor mounting within a triad is actually unnecessary because the equivalent orthogonal sensor signals can be calculated in a system computer from three non-orthogonal (skewed) input axis aligned sensors. Thus, any failed sensor within a triad can be replaced by a single skew aligned sensor, the computer then analytically reforming a new orthogonal triad from the skewed sensor and the two un-failed sensors in the original triad. This basic concept allows each additional level of redundancy to be achieved by addition of one skew aligned sensor rather than duplication of sets of three (i.e., 4 sensors for dual redundancy, 5 sensors for triple redundancy, 6 sensors for quad redundancy). (An additional requirement is that sensor triads formed from the skewed redundant set have non-coplanar input axes.)

For redundant systems in general, means must be provided to detect failures within the redundant set. For skewed strapdown systems, redundancy analysis in the past was primarily focused on identifying sensor failures [1, 2, 4-6], rather than navigation solution failures (as with gimbaled systems). The method was based on monitoring the outputs of "parity" equations, each formed from a linear combination of four or more skewed sensor outputs within the redundant sensor array (and with separate parity equations for skewed redundant gyros and accelerometers). Parity being approximately zero identified that all sensors within the parity group were operating properly (within the expected "normal" range for "failure-free" sensors). Sensors in failure-free identified parity groups were then linearly combined in a weighting algorithm to calculate the equivalent orthogonal axis outputs. This is an effective method for sorting "hard" failures for rapid corrective action of outputs used for flight control. However, for "soft" failures (small errors that would build up over time in an inertial navigation solution), sensor-level parity type testing becomes problematic.

A potential solution to soft failure detection at the sensor parity level is to filter the parity signals so that over time, soft steady failures can be discriminated from normal sensor noise. The basic problem with this approach is that during the time before the failure is detected, the associated sensor error will be building in the navigation solution. Thus, when the sensor causing the error is finally identified (and no longer used for navigation computation input), the navigation error it has already created will remain. To remedy the problem, [3] proposed that first order sensor-to-navigation error sensitivities be generated in parallel with the sensor failure detection parity routines. When the failed sensor (and its error) is finally identified by parity tests, the sensitivity formulas would then be used for "retro-active" correction of the navigation solution error. While this method improves on the previous approach, it was also then recognized that a more proper and direct method for detecting/correcting soft failures in the navigation function was to generate multiple (parallel) inertial navigation solutions using different combinations of redundant sensor inputs. Comparing navigation solutions could then continuously determine those without failures, with the failure-free navigation solutions then used in appropriate weighting formulas for output (in the same manner as with redundant gimbaled INSs). Unfortunately, computer throughput limitations at the time did not permit such a “brute force” parallel navigation solution approach. With continuing advances in computer through-put/memory-size technology, the “brute force” approach to skewed sensor redundancy management is now very feasible. This article describes how such an approach might be implemented.

The article first reviews the classical skewed sensor parity equation method for failure detection. An equivalent parallel strapdown inertial navigation solution approach is then described, each solution formed from a triad of skewed gyro/accelerometer sensor sets within the skew redundant sensor array (gyro and accelerometer input axes are assumed for the article to be parallel). By comparing navigation solutions generated from two of the triads, a navigation parity vector is formed. (Note - A "navigation parity vector" consists of a group of attitude, velocity, and/or position parameter comparisons.) The dual triads used to generate each navigation parity vector are selected from a 4-sensor (tetrad) grouping within the skewed sensor array. The article demonstrates that a navigation parity vector formed from a particular sensor tetrad is equivalent to the traditional skewed sensor parity equation for the tetrad, propagated through linearized error state dynamics of the navigation solution.

A navigation parity vector is generated for each of the possible tetrads within the overall skewed sensor array - e.g., for a skewed 6-axis (hexad) sensor array, there are 15 possible sensor tetrads (15 combinations of 4 sensors within the 6), hence, there would be 15 corresponding navigation parity vectors, each formed by comparing navigation solutions generated from two triads within each tetrad. When all elements of a navigation vector lie within acceptable performance limits, the dual navigation solution inputs forming the parity vector are deemed failure-free, hence, usable for navigation parameter output generation. Conversely, if any element within a navigation parity vector fails its performance limit test, both triad generated navigation solution inputs are deemed failure-suspect, hence, no longer usable for navigation parameter output generation. Navigation outputs would be formed as a weighted average of identified failure-free navigation solution parameters (similar to the gimbaled INS redundancy management approach).

A hexad sensor array configuration is then analyzed in detail. There are 20 possible triads in a hexad for generating parallel navigation solutions (20 combinations of 3 sensors within the 6). The article demonstrates that 12 triads and associated navigation solutions are sufficient to generate the 15 navigation parity vectors that correspond to the 15 possible tetrads. The article then shows that monitoring the 15 navigation parity vectors allows detection/isolation of the first two navigation solution failures, and detection of a third.

NOTATION

$\underline{(\;)}$ = Designation for a vector in general or a column matrix containing the vector components, depending on usage.

$(\underline{V}\times)$ = Skew symmetric (or cross-product) form of vector $\underline{V}$ represented by the square matrix $\begin{bmatrix} 0 & -V_Z & V_Y \\ V_Z & 0 & -V_X \\ -V_Y & V_X & 0 \end{bmatrix}$ in which V_X, V_Y, V_Z are the components of $\underline{V}$. The matrix product of $(\underline{V}\times)$ with another vector equals the cross-product of $(\underline{V}\times)$ with the other vector.

T = Superscript designation for transform of the associated matrix.

$\underline{(\ )}^A$ = Designation for vector $\underline{(\ )}$ represented as a column matrix with elements equal to the projection of $\underline{(\ )}$ on coordinate frame A axes.

TRADITIONAL PARITY EQUATION FAILURE DETECTION APPROACH

Consider an array of n skewed strapdown inertial sensors (n gyros and n accelerometers). The sensor outputs are related to sensor inputs as follows:

$$\underline{s}_{Out} = D\, \underline{s}_{In} + \delta\underline{s} \tag{1}$$

where

$\underline{s}_{Out}$ = Column vector of length n whose elements are the skewed sensor outputs (gyro or accelerometer).

$\underline{s}_{In}$ = Three element sensor input vector whose components equal the projections of strapdown gyro sensed angular rate or accelerometer sensed specific force on orthogonal inertial sensor assembly axes.

$\delta\underline{s}$ = Column vector of length n whose elements are the errors in $\underline{s}_{Out}$.

D = n by 3 geometry matrix. For this article we will assume that the skewed accelerometer and gyro input axes are parallel, hence, D is the same when (1) represents a gyro or an accelerometer output data vector $\underline{s}$.

In general, the elements of $\underline{s}_{Out}$, $\delta\underline{s}$ and D are given by:

$$\begin{aligned} \underline{s}_{Out} &= \left(s_{Out_a}, s_{Out_b}, s_{Out_c}, s_{Out_d}, \cdots s_{Out_n}\right)^T \\ \delta\underline{s} &\equiv \left(\delta s_a, \delta s_b, \delta s_c, \delta s_d, \cdots \delta s_n\right)^T \\ D &= \left(\underline{u}_a, \underline{u}_b, \underline{u}_c, \underline{u}_d, \cdots \underline{u}_n\right)^T \end{aligned} \tag{2}$$

where

s_{Out_i}, δs_i = Output and error of sensor i.

$\underline{u}_i$ = Unit vector parallel to the sensor i input axis.

a, b, c, d, $\cdots$ n = i index designation for sensor a, b, c, d, $\cdots$ n.

A skewed sensor parity vector is defined from Eq. (1) as in [5]:

$$\underline{p} = E\, \underline{s}_{Out} \tag{3}$$

where

$\underline{p}$ = Sensor arity vector in general.

E = Constant matrix.

With (1) in (3),

$$\underline{p} = E\left(D\, \underline{s}_{In} + \delta\underline{s}\right) = E\, D\, \underline{s}_{In} + E\, \delta\underline{s} \tag{4}$$

E is chosen to satisfy the condition:

$$E\, D = 0 \tag{5}$$

Using (5) we see then from (4) that

$$\underline{p} = E\, \delta\underline{s} \tag{6}$$

Thus, $\underline{p}$ measures the errors in $\underline{s}_{Out}$.

For ideal sensors containing zero error, the elements of $\underline{p}$ will all be zero. For non-ideal sensors containing errors under normal sensor operating conditions, the elements of $\underline{p}$ will have small values corresponding to the expected normal sensor errors. Under sensor failure conditions, particular elements of $\underline{p}$ will have large values generated by the larger than normal sensor errors. Those elements of $\underline{p}$ that are affected by particular sensor failures are determined by the rows of E containing non-zero elements in the failed sensor columns. Suitable logic can be designed based on the previous characteristics that enables sensor failures to be detected and isolated. In general, for n skewed sensors, n-3 failures can be detected and n-2 failures can be isolated by monitoring the $\underline{p}$ element behavior patterns.

In the traditional skewed sensor redundancy approach, $\underline{p}$ is monitored as the means for detecting and isolating failed sensors. Remaining un-failed sensors are then used through a suitable weighting matrix to estimate the equivalent orthogonal sensor data. The orthogonal sensor data so generated then forms the input to inertial navigation integration routines that calculate attitude, velocity, and position (and to flight control functions using angular rate and specific force inputs).

SKEWED REDUNDANCY USING PARALLEL NAVIGATION SOLUTIONS

For the parallel navigation solution skewed redundancy approach, individual orthogonal sensor signals are first derived from individual skewed 3-axis (triad) gyro/accelerometer data. The equivalent to a skewed sensor tetrad parity equation is then generated by comparing navigation solutions generated from two (dual) skewed sensor triads, each formed from sensors in a particular tetrad group within the overall redundant sensor array. As such, the triads will have two skewed sensors in common (e.g., for a tetrad with skewed sensors a, b, c and d, if one triad uses sensors a, b, c and the other triad uses sensors b, c, d, sensors b and c will be common between the triads). The navigation parity vector formed by comparing the triad generated navigation solutions represents the equivalent to a traditional sensor parity equation formed from the abcd sensor outputs.

Forming Two Triads Within A Tetrad

For the abc and bcd sensor triads we can write from the appropriate rows of (1):

$$\underline{s}_{Out_{abc}} = D_{abc}\, \underline{s}_{In} + \delta\underline{s}_{abc} \qquad \underline{s}_{Out_{bcd}} = D_{bcd}\, \underline{s}_{In} + \delta\underline{s}_{bcd} \tag{7}$$

with

$$\begin{aligned} &\underline{s}_{Out_{abc}} \equiv \left(s_{Out_a}, s_{Out_b}, s_{Out_c}\right)^T \quad \delta\underline{s}_{abc} \equiv \left(\delta s_a, \delta s_b, \delta s_c\right)^T \quad D_{abc} \equiv \left(\underline{u}_a, \underline{u}_b, \underline{u}_c\right)^T \\ &\underline{s}_{Out_{bcd}} \equiv \left(s_{Out_b}, s_{Out_c}, s_{Out_d}\right)^T \quad \delta\underline{s}_{bcd} \equiv \left(\delta s_b, \delta s_c, \delta s_d\right)^T \quad D_{bcd} \equiv \left(\underline{u}_b, \underline{u}_c, \underline{u}_d\right)^T \end{aligned} \tag{8}$$

Based on (7) we now define:

$$\hat{\underline{s}}_{In_{abc}} \equiv D_{abc}^{-1}\, \underline{s}_{Out_{abc}} \qquad \hat{\underline{s}}_{In_{bcd}} \equiv D_{bcd}^{-1}\, \underline{s}_{Out_{bcd}} \tag{9}$$

which with (7), finds:

$$\hat{\underline{s}}_{In_{abc}} = \underline{s}_{In} + D_{abc}^{-1}\, \delta\underline{s}_{abc} \qquad \hat{\underline{s}}_{In_{bcd}} = \underline{s}_{In} + D_{bcd}^{-1}\, \delta\underline{s}_{bcd} \tag{10}$$

where

$\hat{\underline{s}}_{In_{abc}}, \hat{\underline{s}}_{In_{bcd}}$ = Estimates for the true $\underline{s}_{In}$ sensor input vector obtained using the abc or bcd sensor triad outputs.

Note from (10) that when the sensor errors are zero, the $\hat{\underline{s}}_{In_{abc}}, \hat{\underline{s}}_{In_{bcd}}$ estimates will equal the true $\underline{s}_{In}$ value which is the basis for the form of Eqs. (9).

Appendix A shows that (9) can also be written as

$$\begin{aligned} \hat{\underline{s}}_{In_{abc}} &= \frac{1}{\alpha_{abc}} \left(\underline{u}_b \times \underline{u}_c, \underline{u}_c \times \underline{u}_a, \underline{u}_a \times \underline{u}_b\right) \underline{s}_{Out_{abc}} \\ \hat{\underline{s}}_{In_{bcd}} &= \frac{1}{\alpha_{bcd}} \left(\underline{u}_c \times \underline{u}_d, \underline{u}_d \times \underline{u}_b, \underline{u}_b \times \underline{u}_c\right) \underline{s}_{Out_{bcd}} \end{aligned} \tag{11}$$

in which

$$\alpha_{klm} \equiv \underline{u}_k \cdot \left(\underline{u}_l \times \underline{u}_m\right) \tag{12}$$

and where

$\underline{u}_k, \underline{u}_l, \underline{u}_m$ = Unit vectors along skewed sensor k, *l*, and m input axes.

α_{klm} = Defined vector product operator between unit vectors along the skewed sensor k, *l*, m input axes.

For (11) to exist, α_{abc} and α_{bcd} must be non-zero which is equivalent to the requirement that no three of the abcd sensor input axes lie in the same plane, the classical requirement for skewed redundant sensors [1-6].

Forming A Navigation Parity Vector From The Dual Navigation Solutions

Dual inertial navigation solutions are obtained from the abc and bcd sensor data set by an integration process using $\hat{\underline{s}}_{In_{abc}}$, $\hat{\underline{s}}_{In_{bcd}}$ for the orthogonal inertial sensor vector inputs (gyro sensed angular rate and accelerometer sensed specific force acceleration). For example, a typical aircraft inertial navigation solution generated using an abc or bcd inertial sensor triad would determine attitude, velocity, and position by integrating the following differential equations [8 Sect.2]:

$$\begin{aligned}
\dot{C}_B^N &= C_B^N \left(\hat{\underline{\omega}}_{In}^B \times\right) - \left[\left(\underline{\omega}_e^N + \underline{\rho}^N\right)\times\right] C_B^N \\
\underline{\rho}^N &= F_C^N \left(\underline{u}_{Up}^N \times \underline{v}^N\right) + \rho_{Up}\, \underline{u}_{Up}^N \\
\dot{\underline{v}}^N &= C_B^N \hat{\underline{a}}_{SF_{In}}^B + \underline{g}_P^N - \left(2\, \underline{\omega}_e^N + \underline{\rho}^N\right) \times \underline{v}^N \\
\dot{C}_N^E &= C_N^E \left(\underline{\rho}^N \times\right) \qquad \dot{h} = \underline{u}_{Up}^N \cdot \underline{v}^N
\end{aligned} \tag{13}$$

where

B = Designation for orthogonal "body" frame sensor assembly coordinates.

N = Designation for locally level navigation coordinate (e.g., wander azimuth [8, Sect. 4.5]).

E = Designation for earth fixed coordinates (e.g., one axis parallel to earth's rotation axis, another axis parallel to the intersection of earth's equatorial plane and the Greenwich reference meridian plane, the third perpendicular to the other two).

C_B^N = Direction cosine matrix that transforms vectors from sensor coordinate frame (B) to the locally level navigation frame (N).

C_N^E = Direction cosine matrix that transforms vectors from locally level navigation coordinates (N) to earth fixed reference coordinates (E).

$\hat{\underline{\omega}}_{In}$ = Sensor assembly inertial angular rate vector in body (B) frame coordinates calculated in (11) from a selected skewed gyro triad, e.g., $\hat{\underline{s}}_{In_{abc}}$ or $\hat{\underline{s}}_{In_{bcd}}$ from $\underline{s}_{Out_{abc}}$ or $\underline{s}_{Out_{bcd}}$ skewed gyro measurements.

$\underline{\omega}_e^N$ = Earth inertial rotation rate vector in navigation frame (N) coordinates (a function of C_N^E and h).

$\underline{\rho}^N$ = Angular rate of the navigation frame relative to the earth in N frame coordinates.

ρ_{Up} = Upward component of $\underline{\rho}^N$ (selected based on the type of navigation frame being used, e.g., azimuth wander or free azimuth [7 Sect. 4.5].

$\underline{u}_{Up}^N$ = Unit vector upward in N frame coordinates. Note that for the N frame defined with the third axis up, $\underline{u}_{Up}^N = (0, 0, 1)^T$.

F_C^N = Curvature matrix (a function of C_N^E and h).

$\underline{v}^N$ = Velocity vector relative to the earth in navigation (N) coordinates.

$\hat{\underline{a}}_{SF_{In}}^B$ = Sensor assembly specific force acceleration vector in body (B) frame coordinates calculated in (11) from a selected skewed accelerometer triad, e.g., $\hat{\underline{s}}_{In_{abc}}$ or $\hat{\underline{s}}_{In_{bcd}}$ from $\underline{s}_{Out_{abc}}$ or $\underline{s}_{Out_{bcd}}$ skewed accelerometer measurements.

$\underline{g}_P^N$ = Plumb-bob gravity acceleration in N frame coordinates (a function of earth mass distribution, C_N^E, $\underline{\omega}_e^N$, and h).

h = Altitude.

The previous equations would also typically include updating algorithms to prevent vertical channel divergence [7 Sect. 4.4.1.2.1].

The error-free form of (13) can also be written in the short-hand state vector form:

$$\dot{\underline{X}} = f\left(\underline{X}, \underline{\omega}_{In}, \underline{a}_{SFIn}\right) \tag{14}$$

where

$\underline{X}$ = True navigation parameter state vector with elements equal to the components of C_B^N, $\underline{v}^N$, C_N^E, and h in (13).

$\underline{\omega}_{In}$, $\underline{a}_{SFIn}$ = True inertial angular rate and specific force acceleration vectors of the strapdown inertial sensor assembly.

Now consider the $\hat{\underline{\omega}}_{In_{abc}}$, $\hat{\underline{\omega}}_{In_{bcd}}$ angular rate vectors (calculated in (11) from the abc and bcd skewed gyro outputs) and the $\hat{\underline{a}}_{SFIn_{abc}}$, $\hat{\underline{a}}_{SFIn_{bcd}}$ specific force vectors (calculated in (11) from the abc and bcd skewed accelerometer outputs). The inertial navigation solution for each is obtained by integrating the following in parallel:

$$\dot{\underline{\hat{X}}}_{abc} = f\left(\underline{\hat{X}}_{abc}, \underline{\hat{\omega}}_{In_{abc}}, \underline{\hat{a}}_{SFIn_{abc}}\right) \qquad \dot{\underline{\hat{X}}}_{bcd} = f\left(\underline{\hat{X}}_{bcd}, \underline{\hat{\omega}}_{In_{bcd}}, \underline{\hat{a}}_{SFIn_{bcd}}\right) \tag{15}$$

$$\underline{\hat{X}}_{abc} = \int_0^t \dot{\underline{\hat{X}}}_{abc}(\tau)\, d\tau \qquad \underline{\hat{X}}_{bcd} = \int_0^t \dot{\underline{\hat{X}}}_{bcd}(\tau)\, d\tau \tag{16}$$

where

$\underline{\hat{X}}_{abc}, \underline{\hat{X}}_{bcd}$ = Navigation parameter vector solutions obtained from the abc and bcd sensor derived inertial sensor inputs.

Note: Prior to initiation of the (16) navigation solution integration process, an initial alignment operation would be performed for each sensor triad using the same sensor signals as in (15). Initial alignment typically takes several minutes to complete [8 Chpt. 6]. The parameters generated at initial alignment completion are used to initialize the $\underline{\hat{X}}_{abc}$, $\underline{\hat{X}}_{bcd}$ parameters in (15). For this discussion, initial alignment operations should be considered part of the navigation process.

In principle, the difference between $\underline{\hat{X}}_{abc}(t)$ and $\underline{\hat{X}}_{bcd}(t)$ in (16) would provide the measure needed to identify the presence of error in the abc and/or bcd navigation solutions. However, the form of $\underline{\hat{X}}$ is generally not suitable for direct comparison of all $\underline{\hat{X}}_{abc}(t)$ versus $\underline{\hat{X}}_{bcd}(t)$ elements, necessitating an intermediate method for navigation solution comparison.

COMPARING NAVIGATION SOLUTIONS GENERATED BY THE SENSOR TRIADS

When (14) represents the typical Eq. (13) aircraft INS form (for example), a method is required for comparing the C_B^N and C_N^E matrices, each of which contains 9 elements that are interrelated by orthogonality/normality constraints inherent within a direction cosine matrix. Additionally, comparison of N frame based parameters between different navigation solutions can be problematic if the N frames are not initialized identically in the triad solutions being compared, e.g., [7 Sects. 6.2.1 & 6.2.2] (e.g., analogous to two gimbaled INSs using wander azimuth navigation coordinates in which the platform heading for each may be initialized differently). (Note: This may not actually be a problem with the skewed redundancy situation being analyzed because the solutions are generated within the same system, presumably using the same navigation/initialization routines.) For generality and as an example, the following E frame based navigation parameter equivalents will be used in this article for triad solution comparisons:

$$\begin{aligned} C_B^E &= C_N^E\, C_B^N \qquad & \underline{v}^E &= C_N^E\, \underline{v}^N \\ \underline{u}_{Up}^E &= C_N^E\, \underline{u}_{Up}^N \qquad & \underline{R}^E &\approx \left(R_e + h\right) \underline{u}_{Up}^E \end{aligned} \tag{17}$$

where

C_B^E = Direction cosine matrix that transforms vectors from B to E frame coordinates.

$\underline{v}^E$ = Velocity relative to the earth in E frame coordinates.

$\underline{R}^E$ = Position vector from earth's center to the INS in E frame coordinates.

R_e = Average radius of the earth.

The approximation for $\underline{R}^E$ in (18) is that earth's shape is approximately spherical (See [7 Eqs. (5.2.2-1) & (5.1-10)] for exact $\underline{R}^E$ equation).

Assuming that the integration algorithms used in (16) maintain C_B^N and C_N^E matrix orthogonality/normality, the effective difference between the abc and bcd triad determined C_B^E solutions can be measured based on a rotation vector formulation [7 Sect. 3.2.2.2]:

$$\hat{C}^E_{B_{abc}} = \hat{C}^E_{N_{abc}} \hat{C}^N_{B_{abc}} \qquad \hat{C}^E_{B_{bcd}} = \hat{C}^E_{N_{bcd}} \hat{C}^N_{B_{bcd}}$$

$$C^E_{B_{abcd}} \equiv \hat{C}^E_{B_{abc}} \left(\hat{C}^E_{B_{bcd}}\right)^T \tag{18}$$

$$\left(\underline{\theta}^E_{abcd}\times\right) \equiv \sin\phi_{abcd} \left(\underline{u}^E_{\phi_{abcd}}\times\right) = \frac{1}{2}\left[C^E_{B_{abcd}} - \left(C^E_{B_{abcd}}\right)^T\right]$$

where

$\hat{C}^E_{N_{abc}}, \hat{C}^N_{B_{abc}}, \hat{C}^E_{N_{bcd}}, \hat{C}^N_{B_{bcd}} =$ C_N^E, C_B^N solution within (17) using the abc and bcd skewed sensor derived values for the equivalent orthogonal sensor inputs.

$C^E_{B_{abcd}}$ = Direction cosine matrix formed from $\hat{C}^E_{B_{abc}}, \hat{C}^E_{B_{bcd}}$ as indicated in (18).

ϕ_{abcd} = Magnitude of the rotation vector equivalent to $C^E_{B_{abcd}}$.

$\underline{u}^E_{\phi_{abcd}}$ = Unit vector in E frame axes parallel to the rotation vector equivalent to $C^E_{B_{abcd}}$.

$\underline{\theta}^E_{abcd}$ = Attitude parity vector in E frame axes representing the difference between the $\hat{C}^E_{B_{abc}}, \hat{C}^E_{B_{bcd}}$ solutions.

Note that without abc and bcd sensor errors, the $\hat{C}^E_{B_{abc}}, \hat{C}^E_{B_{bcd}}$ solutions will be identical, hence, $C^E_{B_{abcd}}$ in (18) will be identity and $\underline{\theta}_{abcd}$ in (18) will be zero. Note also that for small ϕ_{abcd}, $\left(\underline{\theta}^E_{abcd}\times\right)$ reduces to $\underline{\theta}^E_{abcd} = \phi_{abcd}\, \underline{u}^E_{\phi_{abcd}} = \underline{\phi}^E_{abcd}$, the rotation vector equivalent to $C^E_{B_{abcd}}$.

The equivalent three component vector form of the $\left(\underline{\theta}^{E}_{abcd}\times\right)$ 3 by 3 matrix in (18) is derived from

$$\underline{u}^{E}_{i} = \underline{u}^{E}_{j} \times \underline{u}^{E}_{k}$$

$$\underline{u}^{E}_{i} \cdot \underline{u}^{E}_{\phi_{abcd}} = \left(\underline{u}^{E}_{j} \times \underline{u}^{E}_{k}\right) \cdot \underline{u}^{E}_{\phi_{abcd}} = \left(\underline{u}^{E}_{\phi_{abcd}} \times \underline{u}^{E}_{j}\right) \cdot \underline{u}^{E}_{k} = \left(\underline{u}^{E}_{k}\right)^{T} \left(\underline{u}^{E}_{\phi_{abcd}} \times\right) \underline{u}^{E}_{j} \tag{19}$$

$$\underline{u}^{E}_{\phi_{abcd}} = \sum_{i=1,3} \left(\underline{u}^{E}_{i} \cdot \underline{u}^{E}_{\phi_{abcd}}\right) \underline{u}^{E}_{i} = \sum_{\substack{ijk = \\ 123,\,231,\,312}} \left[\left(\underline{u}^{E}_{k}\right)^{T} \left(\underline{u}^{E}_{\phi_{abcd}} \times\right) \underline{u}^{E}_{j}\right] \underline{u}^{E}_{i}$$

where

$\underline{u}^{E}_{i},\ \underline{u}^{E}_{j},\ \underline{u}^{E}_{k}$ = Unit vectors parallel to E frame coordinate axes.

Thus, from (18) with (19),

$$\begin{aligned}\underline{\theta}^{E}_{abcd} &= \sin\phi_{abcd}\, \underline{u}^{E}_{\phi_{abcd}} \\ &= \sum_{i=1,3} \left[\underline{u}^{E}_{i} \cdot \left(\sin\phi_{abcd}\, \underline{u}^{E}_{\phi_{abcd}}\right)\right] \underline{u}^{E}_{i} \\ &= \sum_{\substack{ijk = \\ 123,\,231,\,312}} \left\{\left(\underline{u}^{E}_{k}\right)^{T} \frac{1}{2}\left[C^{E}_{B_{abcd}} - \left(C^{E}_{B_{abcd}}\right)^{T}\right] \underline{u}^{E}_{j}\right\} \underline{u}^{E}_{i}\end{aligned} \tag{20}$$

For the particular aircraft INS example problem being analyzed, the accuracy requirements on attitude are generally different for vertical and horizontal components. For accuracy/failure assessment, the vertical/horizontal components of $\underline{\theta}^{E}_{abcd}$ are definable in the N frame as:

$$\begin{aligned}&C^{N}_{E} \approx \frac{1}{2}\left(\hat{C}^{N}_{E_{abc}} + \hat{C}^{N}_{E_{bcd}}\right) \qquad \underline{\theta}^{N}_{abcd} = C^{N}_{E}\, \underline{\theta}^{E}_{abcd} \\ &\theta_{Up_{abcd}} = \underline{\theta}^{N}_{abcd} \cdot \underline{u}^{N}_{Up} \qquad \underline{\theta}^{N}_{H_{abcd}} = \underline{\theta}^{N}_{abcd} - \theta_{Up_{abcd}}\, \underline{u}^{N}_{Up}\end{aligned} \tag{21}$$

where

$\theta_{Up_{abcd}}, \underline{\theta}^{N}_{H_{abcd}}$ = Vertical and horizontal components of $\underline{\theta}^{N}_{abcd}$ in N frame coordinates.

Eqs. (21) with (20) for $\theta_{Up_{abcd}}, \underline{\theta}^{N}_{H_{abcd}}$ would then be used to assess "normal" attitude performance from the abc and bcd triad navigation solutions, as compared with expected horizontal and vertical attitude errors.

The velocity performance of the abc and bcd triads is assessed by first comparing the E frame components:

$$\Delta\underline{v}^{E}_{abcd} = \hat{\underline{v}}^{E}_{abc} - \hat{\underline{v}}^{E}_{bcd} \tag{22}$$

where

$\hat{\underline{v}}^E_{abc}$, $\hat{\underline{v}}^E_{bcd}$ = E frame components of the velocity solutions obtained from the abc and bcd triads.

$\Delta\underline{v}^E_{abcd}$ = Velocity parity vector in the E frame for the abc and bcd triad navigation solutions.

Note that without abc and bcd sensor errors, the $\hat{\underline{v}}^E_{abc}$, $\hat{\underline{v}}^E_{bcd}$ solutions will be identical, hence, $\Delta\underline{v}^E_{abcd}$ in (22) will be zero. For the example aircraft INS, horizontal and vertical velocity accuracy requirements are different, hence, as with attitude accuracy assessment,

$$\Delta\underline{v}^N_{abcd} = C^N_E \, \Delta\underline{v}^E_{abcd} \qquad \Delta v_{Up_{abcd}} = \Delta\underline{v}^N_{abcd} \cdot \underline{u}^N_{Up} \qquad \Delta\underline{v}^N_{H_{abcd}} = \Delta\underline{v}^N_{abcd} - \Delta v_{Up_{abcd}} \, \underline{u}^N_{Up} \tag{23}$$

in which C^N_E is as calculated in (21) and where

$\Delta v_{Up_{abcd}}$, $\Delta\underline{v}^N_{H_{abcd}}$ = Vertical and horizontal components of $\Delta\underline{v}^N_{abcd}$.

Positioning assessment is derived from the difference in abc and bcd triad positioning solutions for C^E_N and h, translated into the equivalent $\underline{R}^E$ with (17):

$$\begin{gathered} \hat{\underline{u}}^E_{Up_{abc}} = \hat{C}^E_{N_{abc}} \, \underline{u}^N_{Up} \qquad \hat{\underline{u}}^E_{Up_{bcd}} = \hat{C}^E_{N_{bcd}} \, \underline{u}^N_{Up} \\ \Delta\underline{R}^E_{abcd} \equiv \hat{\underline{R}}^E_{abc} - \hat{\underline{R}}^E_{bcd} = \left(R_e + \hat{h}_{abc}\right) \hat{\underline{u}}^E_{Up_{abc}} - \left(R_e + \hat{h}_{bcd}\right) \hat{\underline{u}}^E_{Up_{bcd}} \end{gathered} \tag{24}$$

where

$\hat{\underline{R}}^E_{abc}$, $\hat{\underline{R}}^E_{bcd}$ = E frame components of the position vector solutions obtained from the abc and bcd triads.

$\Delta\underline{R}^E_{abcd}$ = Position parity vector in the E frame for the abc and bcd triad navigation solutions.

Note that without abc and bcd sensor errors, the $\hat{\underline{R}}^E_{abc}$, $\hat{\underline{R}}^E_{bcd}$ solutions will be identical, hence, $\Delta\underline{R}^E_{abcd}$ in (24) will be zero. For the aircraft INS, horizontal and vertical positioning accuracy requirements are different, hence, as with attitude and velocity accuracy assessment,

$$\begin{gathered} \Delta\underline{R}^N_{abcd} = C^N_E \, \Delta\underline{R}^E_{abcd} \\ \Delta R_{Up_{abcd}} = \Delta\underline{R}^N_{abcd} \cdot \underline{u}^N_{Up} \qquad \Delta\underline{R}^N_{H_{abcd}} = \Delta\underline{R}^N_{abcd} - \Delta R_{Up_{abcd}} \, \underline{u}^N_{Up} \end{gathered} \tag{25}$$

in which C^N_E is as calculated in (21) and where

$\Delta R_{Up_{abcd}}, \Delta \underline{R}^N_{H_{abcd}}$ = Vertical and horizontal components of $\Delta \underline{R}^N_{abcd}$.

Eqs. (20) - (25) allow the difference between abc and bcd attitude/velocity/position solutions from (13) - (16) (i.e., 9 elements of C^N_B, 9 elements of C^E_N, 3 elements of $\underline{v}^N$, and 1 element of h - 22 elements in all), to be expressed by 9 elements: $\theta_{Up_{abcd}}, \Delta v_{Up_{abcd}}, \Delta R_{Up_{abcd}}$ and 2 elements in each of $\underline{\theta}^N_{H_{abcd}}, \Delta \underline{v}^N_{H_{abcd}}, \Delta \underline{R}^N_{H_{abcd}}$. Thus, the composite navigation parity vector would be

$$\Delta \underline{\zeta}_{abcd} \equiv \left[\left(\underline{\theta}^N_{H_{abcd}} \right)^T, \theta_{Up_{abcd}}, \left(\Delta \underline{v}^N_{H_{abcd}} \right)^T, \Delta v_{Up_{abcd}}, \left(\Delta \underline{R}^N_{H_{abcd}} \right)^T, \Delta R_{Up_{abcd}} \right]^T \quad (26)$$

where

$\Delta \underline{\zeta}_{abcd}$ = Nine element navigation parity vector representing the difference between the abc and bcd triad navigation solutions to (13) - (16).

For the (15) - (16) navigation equation set representing the integral solutions of (13) to abc and bcd sensor triad inputs, (26) would be used as the measure of how closely $\hat{\underline{X}}_{abc}$ and $\hat{\underline{X}}_{bcd}$ compare (i.e., versus $\underline{\theta}^N_{H_{abcd}}, \theta_{Up_{abcd}}, \Delta \underline{v}^N_{H_{abcd}}, \Delta v_{Up_{abcd}}, \Delta \underline{R}^N_{H_{abcd}}, \Delta R_{Up_{abcd}}$ navigation parameter accuracy requirements). If all $\Delta \underline{\zeta}_{abcd}$ component error measurements are deemed acceptable, both $\hat{\underline{X}}_{abc}$ and $\hat{\underline{X}}_{bcd}$ would be treated as valid, allowing their components to be used with appropriate averaging for navigation system parameter outputs. For systems with more than 4 skewed axis sensors, multiple triad navigation solutions would be generated and compared by twos in the same manner. The final navigation parameter outputs would be generated by averaging the outputs from all triads deemed acceptable.

EQUIVALENCY BETWEEN SKEWED REDUNDANT SENSOR AND PARALLEL NAVIGATION PARITY APPROACHES

The previous developments showed how for a (13) type set of navigation equations, $\Delta \underline{\zeta}_{abcd}$ in (26) can be used as the measure of triad abc and bcd navigation parameter accuracy for failure detection. For a navigation equation set different than (13), an equivalent formula can be derived for $\Delta \underline{\zeta}_{abcd}$. We could have also have defined navigation parity as simply the difference between navigation solution parameters generated directly with the abc and bcd sensor triads, e.g., Eqs. (13) - (16). For both definitions, there is a corresponding parity equation that can be formed from the abcd sensors. This section derives equivalency relationships between the navigation parity vector and sensor parity equation for both navigation parity definitions.

Navigation/Sensor Parity Equivalency For Directly Generated Navigation Parameters

For navigation parity defined as the direct difference between the abc and bcd triad $\underline{X}$ navigation solutions:

$$\Delta\underline{X}_{abcd} \equiv \hat{\underline{X}}_{abc} - \hat{\underline{X}}_{bcd} \tag{27}$$

where

$\Delta\underline{X}_{abcd}$ = Navigation solution parity vector that equals zero when the abc and bcd triads are error free.

The sensor induced errors in $\hat{\underline{X}}_{abc}$, $\hat{\underline{X}}_{bcd}$ are defined as variations from the true navigation solution:

$$\hat{\underline{X}}_{abc} = \underline{X} + \delta\underline{X}_{abc} \qquad \hat{\underline{X}}_{bcd} = \underline{X} + \delta\underline{X}_{bcd} \tag{28}$$

where

$\underline{X}$ = True (error free) navigation solution.

$\delta\underline{X}_{abc}$, $\delta\underline{X}_{bcd}$ = Errors in the $\hat{\underline{X}}_{abc}$, $\hat{\underline{X}}_{bcd}$ navigation solutions caused by sensor errors.

The $\delta\underline{X}_{abc}$, $\delta\underline{X}_{bcd}$ navigation errors can be approximated by the following linearized forms [7 Sect. 12.2.1, 12.2.2, & 12.2.3]:

$$\delta\underline{X}_{abc} \approx M(t)\,\underline{x}_{abc} \qquad \delta\underline{X}_{bcd} \approx M(t)\,\underline{x}_{bcd} \tag{29}$$

where

$\underline{x}_{abc}$, $\underline{x}_{bcd}$ = Error state vectors associated with $\hat{\underline{X}}_{abc}$, $\hat{\underline{X}}_{bcd}$.

M(t) = Matrix for converting $\underline{x}_{abc}$, $\underline{x}_{bcd}$ to $\delta\underline{X}_{abc}$, $\delta\underline{X}_{bcd}$. M(t) is a function of $\underline{X}$, hence, varies with time.

The $\underline{x}_{abc}$ and $\underline{x}_{bcd}$ error state vectors are what is typically used for linearized inertial navigation system error analysis [9 Sect. 5.1]. In general, $\underline{x}_{abc}$ and $\underline{x}_{bcd}$ contain 9 elements each (3 attitude error states, 3 velocity error states, 3 position error states - e.g., $\underline{\psi}$, $\delta\underline{V}$, $\delta\underline{R}$ in [7 Sect. 12.3.3 & 8 Eqs. (46)] that are analytically related to the $\delta\underline{X}_{abc}$, $\delta\underline{X}_{bcd}$ navigation parameter error vectors which, for the example given in (13), have 22 elements each. Similarly for $\underline{x}_{bcd}$ versus $\delta\underline{X}_{bcd}$.

Based on (26) - (28), we can also define a linearized navigation parity vector as the difference between the $\underline{x}_{abc}$ and $\underline{x}_{bcd}$ error state vectors:

$$\Delta\underline{x}_{abcd} \equiv \underline{x}_{abc} - \underline{x}_{bcd} \tag{30}$$

which with (27) - (29) shows that

$$\Delta \underline{X}_{abcd} = M(t)\, \Delta \underline{x}_{abcd} \tag{31}$$

For the navigation errors created by gyro or accelerometer sensor error ($\delta \underline{s}_{In_{abc}}$ and $\delta \underline{s}_{In_{bcd}}$), the $\underline{x}_{abc}$ and $\underline{x}_{bcd}$ error state vectors equal the integral of the following classical forms [7 Sect. 15.1, 8 Eqs. (51), 9 Sect. 5.1]:

$$\begin{aligned} \dot{\underline{x}}_{abc}(t) &= A_x(t)\, \underline{x}_{abc}(t) + A_{\delta s}(t)\, \delta \underline{s}_{In_{abc}}(t) \\ \dot{\underline{x}}_{bcd}(t) &= A_x(t)\, \underline{x}_{bcd}(t) + A_{\delta s}(t)\, \delta \underline{s}_{In_{bcd}}(t) \end{aligned} \tag{32}$$

where

$A_x(t)$, $A_{\delta s}(t)$ = Navigation error state dynamic matrix partitions (functions of $\underline{X}$) coupling $\underline{x}$ navigation and $\delta \underline{s}$ inertial sensor errors (gyro or accelerometer), into the $\underline{x}$ navigation error states.

From (10) it should be clear that

$$\delta \underline{s}_{In_{abc}} = D_{abc}^{-1}\, \delta \underline{s}_{Out_{abc}} \qquad \delta \underline{s}_{In_{bcd}} = D_{bcd}^{-1}\, \delta \underline{s}_{Out_{bcd}} \tag{33}$$

Taking the derivative of (30) and substituting (32) - (33) then gives

$$\Delta \dot{\underline{x}}_{abcd}(t) = A_x(t)\, \Delta \underline{x}_{abcd}(t) + A_{\delta s}(t) \left(D_{abc}^{-1}\, \delta \underline{s}_{abc}(t) - D_{bcd}^{-1}\, \delta \underline{s}_{bcd}(t) \right) \tag{34}$$

Integrating (34) obtains $\Delta \underline{x}_{abcd}$ which, through (31), provides the linearized version of navigation parameter parity vector $\Delta \underline{X}_{abcd}$ as a function of abcd sensor errors.

Let us now consider a particular form for a sensor parity vector based on directly comparing orthogonal sensor vectors derived from the abc and bcd sensors:

$$\underline{p}_{abcd} \equiv \hat{\underline{s}}_{In_{abc}} - \hat{\underline{s}}_{In_{bcd}} \tag{35}$$

where

$\underline{p}_{abcd}$ = Sensor parity vector formed from skewed inertial sensor (gyro or accelerometer) abcd outputs.

With (10), (35) is

$$\underline{p}_{abcd} = D_{abc}^{-1}\, \delta \underline{s}_{abc} - D_{bcd}^{-1}\, \delta \underline{s}_{bcd} \tag{36}$$

Comparing (36) with (6), shows that $\underline{p}_{abcd}$ satisfies the traditional sensor parity vector definition.

Eq. (A-17) in Appendix A show that $\underline{p}_{abcd}$ in (36) can also be written as:

$$\underline{p}_{abcd} = \frac{P_{abcd}}{\alpha_{abc}\,\alpha_{bcd}} \sum_{j=1,3} \alpha_{jbc}\,\underline{u}_j \tag{37}$$

in which

$$\alpha_{jlm} \equiv \underline{u}_j \cdot \left(\underline{u}_l \times \underline{u}_m\right) \qquad \alpha_{klm} \equiv \underline{u}_k \cdot \left(\underline{u}_l \times \underline{u}_m\right)$$
$$P_{abcd} \equiv \alpha_{bcd}\, s_{Out_a} - \alpha_{cda}\, s_{Out_b} + \alpha_{dab}\, s_{Out_c} - \alpha_{abc}\, s_{Out_d} \tag{38}$$

and where

$\underline{u}_j$ = Unit vector along the sensor assembly j coordinate axis. Sensor assembly coordinates are defined as an orthogonal right hand set fixed to the sensor assembly.

$\underline{u}_k$, $\underline{u}_l$, $\underline{u}_m$ = Unit vectors along skewed sensor k, *l*, and m input axes.

P_{abcd} = Scalar sensor parity equation for the abcd sensors (i.e., equals zero under error free abcd sensor conditions, and measures abcd sensor errors otherwise).

Substituting (37) for (36) in (34) gives

$$\Delta\dot{\underline{x}}_{abcd}(t) = A_x(t)\,\Delta\underline{x}_{abcd}(t) + A_{\delta s}(t)\,\frac{P_{abcd}}{\alpha_{abc}\,\alpha_{bcd}} \sum_{j=1,3} \alpha_{jbc}\,\underline{u}_j \tag{39}$$

whose integral is

$$\Delta\underline{x}_{abcd}(t) = \int_0^t \Delta\dot{\underline{x}}_{abcd}(\tau)\,d\tau \tag{40}$$

or with (31),

$$\Delta\underline{X}_{abcd}(t) = M(t) \int_0^t \Delta\dot{\underline{x}}_{abcd}(\tau)\,d\tau \tag{41}$$

Thus, from (39) - (41) we see that on a linearized basis, the navigation parameter parity vector $\Delta\underline{X}_{abcd}$ is proportional to the propagated P_{abcd} scalar sensor parity equation for the abcd sensors, propagated through the navigation error state dynamic equations.

Navigation/Sensor Parity Equivalency For Converted Navigation Parameters

For a $\Delta\underline{\zeta}_{abcd}$ type navigation parity vector representing the difference between abc and bcd triad generated navigation solutions (e.g., Eq. (26) for the Eqs. (13) aircraft INS), the equivalency between $\Delta\underline{\zeta}_{abcd}$ and the corresponding abcd parity equation is more directly defined. For example, similar to the derivation leading to (26), the $\underline{x}_{abc}$, $\underline{x}_{bcd}$ error state vector elements might have been derived by first defining the errors to be in the computed Eq. (17) E frame equivalent navigation parameters, then transforming the E frame defined navigation errors into the N frame for horizontal/vertical component extraction [7 Sects. 12.2.1 - 12.2.3, 8 Eqs. (46)]. The derivation would closely match (18) - (26), except that the final result, compared with (26), would be for the individual abc and bcd triad generated navigation solutions (not the (18) - (26) defined parity difference):

$$\begin{aligned} \underline{x}_{abc} &\equiv \left[\left(\underline{\psi}^N_{H_{abc}}\right)^T, \psi_{Up_{abc}}, \left(\delta\underline{V}^N_{H_{abc}}\right)^T, \delta V_{Up_{abc}}, \left(\delta\underline{R}^N_{H_{abc}}\right)^T, \delta R_{Up_{abc}} \right]^T \\ \underline{x}_{bcd} &\equiv \left[\left(\underline{\psi}^N_{H_{bcd}}\right)^T, \psi_{Up_{bcd}}, \left(\delta\underline{V}^N_{H_{bcd}}\right)^T, \delta V_{Up_{bcd}}, \left(\delta\underline{R}^N_{H_{bcd}}\right)^T, \delta R_{Up_{bcd}} \right]^T \end{aligned} \qquad (42)$$

where

$\underline{\psi}^N_H$, ψ_{Up} = Horizontal and vertical (up) components of the small rotation angle error in the computed $\hat{C}^E_B$ attitude matrix (nominally defined in (17)), define in the E frame then transformed to the N frame.

$\delta\underline{V}^N_H$, δV_{Up}, $\delta\underline{R}^N_H$, δR_{Up} = Horizontal and vertical components of the errors in computed $\hat{\underline{v}}^E$, $\hat{\underline{R}}^E$ (nominally defined in (17)), defined in the E frame then transformed to the N frame.

Eq. (26) would be rederived, beginning with a linearized version of the error in the computed (17) navigation parameters generated using each of the abc and bcd triads:

$$\begin{gathered} \delta C^E_B \equiv \hat{C}^E_B - C^E_B = - \left(\underline{\psi}^E \times\right) \hat{C}^E_B \qquad \delta\underline{V}^E \equiv \hat{\underline{v}}^E - \underline{v}^E \qquad \delta\underline{R}^E \equiv \hat{\underline{R}}^E - \underline{R}^E \\ \underline{\zeta}^E \equiv \left[\left(\underline{\psi}^E\right)^T, \left(\delta\underline{V}^E\right)^T, \left(\delta\underline{R}^E\right)^T \right]^T \qquad \underline{\zeta}^N = \hat{C}^N_E \, \underline{\zeta}^E \end{gathered} \qquad (43)$$

where

$\hat{C}^E_B$, $\hat{\underline{v}}^E$, $\hat{\underline{R}}^E$ = Values for the (17) navigation parameters calculated with sensor inputs from each of the abc and bcd triads.

C^E_B, $\underline{v}^E$, $\underline{R}^E$ = Nominal (error free) values for $\hat{C}^E_B$, $\hat{\underline{v}}^E$, $\hat{\underline{R}}^E$.

δC^E_B, $\delta\underline{V}^E$, $\delta\underline{R}^E$ = Attitude, velocity, position errors defined in the E frame.

$\underline{\psi}^E$ = Small angle rotation vector equivalent to δC^E_B , see [7 Sect. 12.2.1].

Then navigation parity vector $\Delta\underline{\zeta}_{abcd}$ would be defined as the difference between the abc and bcd triad generated $\underline{\zeta}^N$s:

$$\Delta\underline{\zeta}_{abcd} = \underline{\zeta}^N_{abc} - \underline{\zeta}^N_{bcd} \tag{44}$$

The (44) result exactly matches $\Delta\underline{x}_{abcd}$ in (30) formed as the difference between the (42) $\underline{x}_{abc}$, $\underline{x}_{bcd}$ vectors. Furthermore, it can be verified that $\Delta\underline{\zeta}_{abcd}$ in (44) exactly matches the linearized form of $\Delta\underline{\zeta}_{abcd}$ in (26) in which individual elements are defined as the difference between abc and bcd triad generated navigation errors:

$$\begin{aligned}
&\underline{\theta}^N_{H_{abcd}} \equiv \underline{\psi}^N_{H_{abc}} - \underline{\psi}^N_{H_{bcd}} \qquad && \theta_{Up_{abcd}} \equiv \psi_{Up_{abc}} - \psi_{Up_{bcd}} \\
&\Delta\underline{v}^N_{H_{abcd}} \equiv \delta\underline{V}^N_{H_{abc}} - \delta\underline{V}^N_{H_{bcd}} \qquad && \Delta v_{Up_{abcd}} \equiv \delta V_{Up_{abc}} - \delta V_{Up_{bcd}} \\
&\Delta\underline{R}^N_{H_{abcd}} \equiv \delta\underline{R}^N_{H_{abc}} - \delta\underline{R}^N_{H_{bcd}} \qquad && \Delta R_{Up_{abcd}} \equiv \delta R_{Up_{abc}} - \delta R_{Up_{bcd}}
\end{aligned} \tag{45}$$

Thus, for $\underline{x}_{abc}$, $\underline{x}_{bcd}$ and $\underline{\zeta}^N_{abc}$, $\underline{\zeta}^N_{bcd}$ error components defined the same, $\underline{\zeta}^N_{abcd} = \Delta\underline{x}_{abcd}$, and

$$\Delta\underline{\zeta}_{abcd} = \int_0^t \Delta\dot{\underline{x}}_{abcd}(\tau)\, d\tau \tag{46}$$

in which $\Delta\dot{\underline{x}}_{abcd}$ is provided by (39). From (39) and (46) we see that on a linearized basis, a $\Delta\underline{\zeta}_{abcd}$ type navigation parity vector is proportional to the propagated P_{abcd} scalar sensor parity equation for the abcd sensors, propagated through the navigation error state dynamic equations.

HEXAD EXAMPLE

As an example of how skewed sensor parallel navigation solution redundancy might be implemented, consider a skewed hexad sensor array configuration (i.e. 6 skewed gyro/accelerometer sets). In a hexad there are 15 potential tetrad (four) skewed sensor sets (i.e., the number of combinations of 4 contained in 6 is 6 ! / [4 ! (6 - 4) !] = 15). For the 1, 2, 3, 4, 5, and 6 skewed hexad sensors, the fifteen tetrad sensor groups are as follows:

Tetrad 56 = 1234
Tetrad 46 = 1235
Tetrad 45 = 1236
Tetrad 36 = 1245
Tetrad 35 = 1246
Tetrad 34 = 1256
Tetrad 26 = 1345 (47)
Tetrad 25 = 1346
Tetrad 24 = 1356
Tetrad 23 = 1456
Tetrad 16 = 2345
Tetrad 15 = 2346
Tetrad 14 = 2356
Tetrad 13 = 2456
Tetrad 12 = 3456

where

Tetrad IJ = Set of 4 of the 6 skewed sensors that does not include sensors I and J.

For each tetrad, two (of the possible four) triads are selected for navigation solution generation, for Tetrad 34 in Eqs. (47) for example, triads 156 and 256. The 156 and 256 skewed triads would be Eq. (11) orthogonalized, each then used to generate a navigation solution. The navigation parity vector for Tetrad 34 would then be obtained by comparing the two triad 156 and 256 generated navigation solutions. Fifteen navigation parity vectors would thereby be defined, one for each of the Eq. (47) sensor tetrads:

Tetrad 56 = 1234 $\Rightarrow \Delta \underline{X}_{56}$
Tetrad 46 = 1235 $\Rightarrow \Delta \underline{X}_{46}$
Tetrad 45 = 1236 $\Rightarrow \Delta \underline{X}_{45}$
Tetrad 36 = 1245 $\Rightarrow \Delta \underline{X}_{36}$
Tetrad 35 = 1246 $\Rightarrow \Delta \underline{X}_{35}$
Tetrad 34 = 1256 $\Rightarrow \Delta \underline{X}_{34}$
Tetrad 26 = 1345 $\Rightarrow \Delta \underline{X}_{26}$ (48)
Tetrad 25 = 1346 $\Rightarrow \Delta \underline{X}_{25}$
Tetrad 24 = 1356 $\Rightarrow \Delta \underline{X}_{24}$
Tetrad 23 = 1456 $\Rightarrow \Delta \underline{X}_{23}$
Tetrad 16 = 2345 $\Rightarrow \Delta \underline{X}_{16}$
Tetrad 15 = 2346 $\Rightarrow \Delta \underline{X}_{15}$
Tetrad 14 = 2356 $\Rightarrow \Delta \underline{X}_{14}$
Tetrad 13 = 2456 $\Rightarrow \Delta \underline{X}_{13}$
Tetrad 12 = 3456 $\Rightarrow \Delta \underline{X}_{12}$

where

$\Delta \underline{X}_{IJ}$ = Navigation parity vector formed by differencing navigation solution parameters generated from the sensor triads selected within each Tetrad IJ (defined as the tetrad that does not contain sensors I and J).

There are 20 potential triads that can be defined in a hexad (i.e., the number of combinations of 3 contained in a hexad is 6 ! / [3 ! (6 - 3) !] = 20). Note - More than one tetrad contains the same triad, e.g., the 123 triad can be found in the 1234, 1235, and 1236 tetrads. This enables less than 20 triad generated navigation solutions to form all 15 possible navigation parity vectors. For example, by analytical experimentation it has been determined that the 15 Eq. (47) tetrads can be formed from the following 12 of 20 possible triads:

$$\hat{\underline{X}}_{123} \quad \hat{\underline{X}}_{134} \quad \hat{\underline{X}}_{145} \quad \hat{\underline{X}}_{146} \quad \hat{\underline{X}}_{156} \quad \hat{\underline{X}}_{235} \quad \hat{\underline{X}}_{236} \quad \hat{\underline{X}}_{245} \quad \hat{\underline{X}}_{246} \quad \hat{\underline{X}}_{256} \quad \hat{\underline{X}}_{345} \quad \hat{\underline{X}}_{356} \qquad (49)$$

where

$\hat{\underline{X}}_{IJK}$ = Navigation parameter vector solution generated using IJK skewed sensor triad derived inertial sensor inputs.

The navigation parameter parity vectors in (48) are formed by differencing navigation solutions in (49) generated using sensor triads within each parity vector's tetrad:

$$\begin{aligned}
\Delta \underline{X}_{56} &= \hat{\underline{X}}_{123} - \hat{\underline{X}}_{134} \\
\Delta \underline{X}_{46} &= \hat{\underline{X}}_{123} - \hat{\underline{X}}_{235} \\
\Delta \underline{X}_{45} &= \hat{\underline{X}}_{123} - \hat{\underline{X}}_{236} \\
\Delta \underline{X}_{36} &= \hat{\underline{X}}_{145} - \hat{\underline{X}}_{245} \\
\Delta \underline{X}_{35} &= \hat{\underline{X}}_{146} - \hat{\underline{X}}_{246} \\
\Delta \underline{X}_{34} &= \hat{\underline{X}}_{156} - \hat{\underline{X}}_{256} \\
\Delta \underline{X}_{26} &= \hat{\underline{X}}_{134} - \hat{\underline{X}}_{145} \\
\Delta \underline{X}_{25} &= \hat{\underline{X}}_{134} - \hat{\underline{X}}_{146} \\
\Delta \underline{X}_{24} &= \hat{\underline{X}}_{156} - \hat{\underline{X}}_{356} \\
\Delta \underline{X}_{23} &= \hat{\underline{X}}_{145} - \hat{\underline{X}}_{156} \\
\Delta \underline{X}_{16} &= \hat{\underline{X}}_{235} - \hat{\underline{X}}_{245} \\
\Delta \underline{X}_{15} &= \hat{\underline{X}}_{236} - \hat{\underline{X}}_{246} \\
\Delta \underline{X}_{14} &= \hat{\underline{X}}_{235} - \hat{\underline{X}}_{236} \\
\Delta \underline{X}_{13} &= \hat{\underline{X}}_{245} - \hat{\underline{X}}_{256} \\
\Delta \underline{X}_{12} &= \hat{\underline{X}}_{345} - \hat{\underline{X}}_{356}
\end{aligned} \qquad (50)$$

Failure detection/isolation logic using the (50) navigation parity equations would be as follows.

For the first sensor failure, 5 of the Eqs. (50) navigation parity vectors would remain within accuracy limits. For example, for a sensor 1 failure, parity vectors $\Delta \underline{X}_{16}$, $\Delta \underline{X}_{15}$, $\Delta \underline{X}_{14}$, $\Delta \underline{X}_{13}$, and $\Delta \underline{X}_{12}$ would remain within accuracy limits because none have used a triad navigation solution derived using sensor 1. The other 10 parity vectors would have exceeded their accuracy limits which, for this example set of 10 (those for which I or J is not 1), identifies sensor 1 as the failed sensor. Following the sensor 1 failure, only the triads that generated the $\Delta\underline{X}_{16}$, $\Delta\underline{X}_{15}$, $\Delta\underline{X}_{14}$, $\Delta\underline{X}_{13}$, $\Delta\underline{X}_{12}$ parity vectors in Eq. (50) would be used to generate navigation outputs (i.e., $\hat{\underline{X}}_{235}$, $\hat{\underline{X}}_{236}$, $\hat{\underline{X}}_{245}$, $\hat{\underline{X}}_{246}$, $\hat{\underline{X}}_{256}$, $\hat{\underline{X}}_{345}$, $\hat{\underline{X}}_{356}$). The $\Delta\underline{X}_{16}$, $\Delta\underline{X}_{15}$, $\Delta\underline{X}_{14}$, $\Delta\underline{X}_{13}$, $\Delta\underline{X}_{12}$ parity vectors would continue to be monitored to detect the next failure.

For the second sensor failure, only one of the remaining parity vectors will continue to meet accuracy requirements (i.e., the $\Delta\underline{X}_{IJ}$ in which I and J correspond to the two failed sensors). The parity vector that continued to satisfy accuracy limits, identifies (isolates) the second failed sensor (i.e., given that I was identified as the first sensor to fail, the second failed sensor would be J in the unfailed $\Delta\underline{X}_{IJ}$ parity vector). After the second sensor failure, the two navigation solutions used in the remaining "failure-free" navigation parity vector would continue to be used to generate navigation outputs. The remaining parity vector would then be monitored to detect another failure.

A third failure will cause the remaining parity vector to exceed accuracy limits, but that failure cannot be isolated to a particular sensor (or triad navigation solution). Consequently, accuracy assured navigation outputs could not be generated following the third detected failure without an alternative navigation device for comparison (e.g., GPS).

For soft failures, the previous patterns may not develop simultaneously for sensor failure identification. However, the triads with identified failure-free navigation parity vectors would still be satisfactory for output generation. For hard or soft failures, the output navigation parameters would be obtained by averaging the navigation parameters from triads that were used in the failure-free navigation parity vectors.

APPENDIX A

EQUIVALENCY BETWEEN A NAVIGATION PARITY VECTOR AND ITS ASSOCIATED TETRAD SENSOR PARITY EQUATION

A sensor parity vector for the abcd sensor tetrad was defined in Eq. (35) as:

$$\underline{p}_{abcd} \equiv \hat{\underline{s}}\mathrm{In}_{abc} - \hat{\underline{s}}\mathrm{In}_{bcd} \qquad \text{(A-1)}$$

Expanding (A-1) along sensor assembly j axes finds:

$$\underline{p}_{abcd} = \sum_{j=1,3} p_{abcd/j}\, \underline{u}_j \qquad \text{(A-2)}$$

in which

$$p_{abcd/j} \equiv \underline{u}_j \cdot \left(\hat{\underline{s}}\mathrm{In}_{abc} - \hat{\underline{s}}\mathrm{In}_{bcd}\right) \qquad \text{(A-3)}$$

where

$p_{abcd/j}$ = Component of $\underline{p}_{abcd}$ along sensor assembly axis j.

$\underline{u}_j$ = Unit vector along the sensor assembly j coordinate axis. Sensor assembly coordinates are defined as an orthogonal right hand set fixed to the sensor assembly.

Using (9), Eq. (A-3) is equivalently:

$$p_{abcd/j} = \underline{u}_j^T \left(D_{abc}^{-1} \underline{sOut}_{abc} - D_{bcd}^{-1} \underline{sOut}_{bcd} \right) \tag{A-4}$$

where

$\underline{V}^T$ = Transpose of the column matrix formed from the components of $\underline{V}$.

For derivations to follow, the following identities will be useful:

$$\begin{aligned}
&\underline{V}_k \cdot \left(\underline{V}_l \times \underline{V}_m\right) = \underline{V}_m \cdot \left(\underline{V}_k \times \underline{V}_l\right) = \underline{V}_l \cdot \left(\underline{V}_m \times \underline{V}_k\right) \\
&\underline{V}_k \cdot \left(\underline{V}_l \times \underline{V}_m\right) = - \underline{V}_k \cdot \left(\underline{V}_m \times \underline{V}_l\right) \\
&\underline{V}_k \times \left(\underline{V}_l \times \underline{V}_m\right) = \underline{V}_l \left(\underline{V}_k \cdot \underline{V}_m\right) - \underline{V}_m \left(\underline{V}_k \cdot \underline{V}_l\right) \\
&\left[\left(\underline{V}_k \times \underline{V}_l\right) \times\right] = \underline{V}_l \, \underline{V}_k^T - \underline{V}_k \, \underline{V}_l^T \\
&\left(\underline{V} \times\right)^T = - \left(\underline{V} \times\right)
\end{aligned} \tag{A-5}$$

From the D_{abc}, D_{bcd} definitions in (8) and the classical analytical equations for the elements of a matrix inverse [10 Chpt. 4 Sect. 17] it can be shown that

$$\begin{aligned}
D_{abc}^{-1} &= \frac{1}{\underline{u}_a \cdot \left(\underline{u}_b \times \underline{u}_c\right)} \left(\underline{u}_b \times \underline{u}_c, \underline{u}_c \times \underline{u}_a, \underline{u}_a \times \underline{u}_b\right) \\
D_{bcd}^{-1} &= \frac{1}{\underline{u}_b \cdot \left(\underline{u}_c \times \underline{u}_d\right)} \left(\underline{u}_c \times \underline{u}_d, \underline{u}_d \times \underline{u}_b, \underline{u}_b \times \underline{u}_c\right)
\end{aligned} \tag{A-6}$$

(With D_{abc}, D_{bcd} from (8), (A-6) can be demonstrated to be correct by analytically showing that $B_{abc}\, B_{abc}^{-1}$ and $B_{bcd}\, B_{bcd}^{-1}$ equal the identity matrix.) Substituting (A-6) in (A-4) with (8) and expanding finds

$$
\begin{aligned}
p_{abcd/j} &= \underline{u}_j^T \frac{1}{\underline{u}_a \cdot (\underline{u}_b \times \underline{u}_c)} (\underline{u}_b \times \underline{u}_c,\ \underline{u}_c \times \underline{u}_a,\ \underline{u}_a \times \underline{u}_b) (s_{Out_a},\ s_{Out_b},\ s_{Out_c})^T \\
&\quad - \underline{u}_j^T \frac{1}{\underline{u}_b \cdot (\underline{u}_c \times \underline{u}_d)} (\underline{u}_c \times \underline{u}_d,\ \underline{u}_d \times \underline{u}_b,\ \underline{u}_b \times \underline{u}_c) (s_{Out_b},\ s_{Out_c},\ s_{Out_d})^T \\
&= \underline{u}_j^T \frac{1}{\underline{u}_a \cdot (\underline{u}_b \times \underline{u}_c)} \left[(\underline{u}_b \times \underline{u}_c)\, s_{Out_a},\ (\underline{u}_c \times \underline{u}_a)\, s_{Out_b},\ (\underline{u}_a \times \underline{u}_b)\, s_{Out_c} \right] \\
&\quad - \underline{u}_j^T \frac{1}{\underline{u}_b \cdot (\underline{u}_c \times \underline{u}_d)} \left[(\underline{u}_c \times \underline{u}_d)\, s_{Out_b},\ (\underline{u}_d \times \underline{u}_b)\, s_{Out_c},\ (\underline{u}_b \times \underline{u}_c)\, s_{Out_d} \right] \\
&= \frac{1}{\underline{u}_a \cdot (\underline{u}_b \times \underline{u}_c)} \left[\underline{u}_j \cdot (\underline{u}_b \times \underline{u}_c)\, s_{Out_a} + \underline{u}_j \cdot (\underline{u}_c \times \underline{u}_a)\, s_{Out_b} + \underline{u}_j \cdot (\underline{u}_a \times \underline{u}_b)\, s_{Out_c} \right] \\
&\quad - \frac{1}{\underline{u}_b \cdot (\underline{u}_c \times \underline{u}_d)} \left[\underline{u}_j \cdot (\underline{u}_c \times \underline{u}_d)\, s_{Out_b} + \underline{u}_j \cdot (\underline{u}_d \times \underline{u}_b)\, s_{Out_c} + \underline{u}_j \cdot (\underline{u}_b \times \underline{u}_c)\, s_{Out_d} \right]
\end{aligned}
\tag{A-7}
$$

For simplicity, we adopt the following notation:

$$
\alpha_{jlm} \equiv \underline{u}_j \cdot (\underline{u}_l \times \underline{u}_m) \qquad \alpha_{klm} \equiv \underline{u}_k \cdot (\underline{u}_l \times \underline{u}_m) \tag{A-8}
$$

where

$\underline{u}_k$, $\underline{u}_l$, $\underline{u}_m$ = Unit vectors along skewed sensor k, *l*, and m input axes.

With (A-8), (A-7) becomes:

$$
\begin{aligned}
p_{abcd/j} &= \frac{1}{\alpha_{abc}} (\alpha_{jbc}\, s_{Out_a} + \alpha_{jca}\, s_{Out_b} + \alpha_{jab}\, s_{Out_c}) \\
&\quad - \frac{1}{\alpha_{bcd}} (\alpha_{jcd}\, s_{Out_b} + \alpha_{jdb}\, s_{Out_c} + \alpha_{jbc}\, s_{Out_d}) \\
&= \frac{\alpha_{jbc}}{\alpha_{abc}} s_{Out_a} + \left(\frac{\alpha_{jca}}{\alpha_{abc}} - \frac{\alpha_{jcd}}{\alpha_{bcd}} \right) s_{Out_b} + \left(\frac{\alpha_{jab}}{\alpha_{abc}} - \frac{\alpha_{jdb}}{\alpha_{bcd}} \right) s_{Out_c} - \frac{\alpha_{jbc}}{\alpha_{bcd}} s_{Out_d} \\
&= \frac{\alpha_{jbc}}{\alpha_{abc}} s_{Out_a} + \frac{(\alpha_{jca}\, \alpha_{bcd} - \alpha_{jcd}\, \alpha_{abc})}{\alpha_{abc}\, \alpha_{bcd}} s_{Out_b} + \frac{(\alpha_{jab}\, \alpha_{bcd} - \alpha_{jdb}\, \alpha_{abc})}{\alpha_{abc}\, \alpha_{bcd}} s_{Out_c} - \frac{\alpha_{jbc}}{\alpha_{bcd}} s_{Out_d}
\end{aligned}
\tag{A-9}
$$

Using (A-5) and (A-8), the first bracketed term in (A-9) simplifies as follows. First,

$$\alpha_{jca}\,\alpha_{bcd} - \alpha_{jcd}\,\alpha_{abc} =$$

$$\left[\underline{u}_j \cdot \left(\underline{u}_c \times \underline{u}_a\right)\right]\left[\underline{u}_b \cdot \left(\underline{u}_c \times \underline{u}_d\right)\right] - \left[\underline{u}_j \cdot \left(\underline{u}_c \times \underline{u}_d\right)\right]\left[\underline{u}_a \cdot \left(\underline{u}_b \times \underline{u}_c\right)\right]$$

$$= \left[\left(\underline{u}_c \times \underline{u}_a\right) \cdot \underline{u}_j\right]\left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_b\right] - \left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_j\right]\left[\left(\underline{u}_c \times \underline{u}_a\right) \cdot \underline{u}_b\right]$$

$$= \left(\underline{u}_c \times \underline{u}_a\right)^T \left\{\underline{u}_j \left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_b\right] - \underline{u}_b \left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_j\right]\right\} \qquad \text{(A-10)}$$

$$= \left(\underline{u}_c \times \underline{u}_a\right)^T \left[\left(\underline{u}_c \times \underline{u}_d\right) \times \left(\underline{u}_j \times \underline{u}_b\right)\right]$$

$$= \left(\underline{u}_c \times \underline{u}_a\right)^T \left[\left(\underline{u}_c \times \underline{u}_d\right) \times\ \right] \left(\underline{u}_j \times \underline{u}_b\right)$$

Then for the term $\left(\underline{u}_c \times \underline{u}_a\right)^T \left[\left(\underline{u}_c \times \underline{u}_d\right) \times\ \right]$ in (A-10),

$$\left(\underline{u}_c \times \underline{u}_a\right)^T \left[\left(\underline{u}_c \times \underline{u}_d\right) \times\ \right] = \left\{\left[\left(\underline{u}_c \times \underline{u}_d\right) \times\ \right]^T \left(\underline{u}_c \times \underline{u}_a\right)\right\}^T$$

$$= -\left[\left(\underline{u}_c \times \underline{u}_d\right) \times \left(\underline{u}_c \times \underline{u}_a\right)\right]^T \qquad \text{(A-11)}$$

$$= -\left\{\underline{u}_c \left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_a\right] - \underline{u}_a \left[\left(\underline{u}_c \times \underline{u}_d\right) \cdot \underline{u}_c\right]\right\}^T$$

$$= -\underline{u}_c^T \left[\underline{u}_a \cdot \left(\underline{u}_c \times \underline{u}_d\right)\right]$$

Substituting (A-11) in (A-10) yields the simplified result

$$\alpha_{jca}\,\alpha_{bcd} - \alpha_{jcd}\,\alpha_{abc} =$$

$$= -\underline{u}_c^T \left[\underline{u}_a \cdot \left(\underline{u}_c \times \underline{u}_d\right)\right] \left(\underline{u}_j \times \underline{u}_b\right)$$

$$= -\left[\underline{u}_a \cdot \left(\underline{u}_c \times \underline{u}_d\right)\right]\left[\underline{u}_c \cdot \left(\underline{u}_j \times \underline{u}_b\right)\right] \qquad \text{(A-12)}$$

$$= -\alpha_{acd}\,\alpha_{cjb} = -\alpha_{cda}\,\alpha_{jbc}$$

Similarly, for the second bracketed term in (A-9):

$$\alpha_{jab}\,\alpha_{bcd} - \alpha_{jdb}\,\alpha_{abc} = \alpha_{dab}\,\alpha_{jbc} \qquad \text{(A-13)}$$

With (A-12) and (A-13), (A-9) simplifies to

$$p_{abcd/j} = \frac{\alpha_{jbc}}{\alpha_{abc}}\, s_{Out_a} - \left(\frac{\alpha_{cda}\,\alpha_{jbc}}{\alpha_{abc}\,\alpha_{bcd}}\right) s_{Out_b} + \left(\frac{\alpha_{dab}\,\alpha_{jbc}}{\alpha_{abc}\,\alpha_{bcd}}\right) s_{Out_c} - \frac{\alpha_{jbc}}{\alpha_{bcd}}\, s_{Out_d}$$

$$= \frac{\alpha_{jbc}}{\alpha_{abc}\,\alpha_{bcd}} \left(\alpha_{bcd}\, s_{Out_a} - \alpha_{cda}\, s_{Out_b} + \alpha_{dab}\, s_{Out_c} - \alpha_{abc}\, s_{Out_d}\right) \qquad \text{(A-14)}$$

Identifying the bracketed term in (A-14) as a scalar abcd sensor parity equation,

$$P_{abcd} \equiv \alpha_{bcd}\, s_{Out_a} - \alpha_{cda}\, s_{Out_b} + \alpha_{dab}\, s_{Out_c} - \alpha_{abc}\, s_{Out_d} \tag{A-15}$$

Eq. (A-14) becomes

$$p_{abcd/j} = \frac{\alpha_{jbc}}{\alpha_{abc}\, \alpha_{bcd}} P_{abcd} \tag{A-16}$$

Eq. (A-16) is then substituted into (A-2) to obtain the final result:

$$\underline{p}_{abcd} = \frac{P_{abcd}}{\alpha_{abc}\, \alpha_{bcd}} \sum_{j=1,3} \alpha_{jbc}\, \underline{u}_j \tag{A-17}$$

REFERENCES

1. Weiss, R. & Nathan, I., "An Ultra-reliable Sensing System for Vehicles with Limited Sparing Capability", AIAA Journal of Spacecraft, Vo. 4, No. 9, September 1967.

2. Gilmore, J. P. & McKern, R. A., "A Redundant Strapdown Inertial Reference Unit (SIRU)", AIAA Journal of Spacecraft, Vol. 9, No. 1, January 1972.

3. Wilcox, J. C., "Retroactive Failure Correction for Strapdown Redundant Inertial Instruments", AIAA Journal of Spacecraft, Vol. 12, No. 6, June 1975.

4. Ebner, R. E. & Mark, J. G., "Redundant Integrated Flight-Control/Navigation Inertial Sensor Complex", AIAA Journal of Guidance and Control, Vol. 1, No. 2, March-April 1978.

5. Hall, S. R.; Motyka, P.; Gai, E.; Deyst, Jr., J. J.; "In-Flight Parity Vector Compensation for FDI", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-19, No. 5 September 1983.

6. Sukkarieh, S.; Gibbens, P.; Grocholsky, B.; Willis, K.; Durrant-Whyte, H.), "A Low-Cost Redundant Inertial Measurement Unit for Unmanned Air Vehicles", The International Journal of Robotics Research, November 2000.

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

8. 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 Free Access as SAI-WBN-14010, May 31, 2015 at wwww.strapdownassociates.com.

9. Savage, P. G., "Performance Analysis Of Strapdown Systems", *Low Cost Navigation Sensors and Integration Technology*, NATO RTO-EN-SET-116(2008), Section 10, published in 2009, also reformatted for Web Free Access as SAI-WBN-14011, June 2, 2015 at wwww.strapdownassociates.com.

10. Sokolnikoff, I. S. & Redheffer, R. M.; *Mathematics of Physics and Modern Engineering*, McGraw-Hill, New York, London, Toronto, 1958.