JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 29, No. 2, March–April 2006

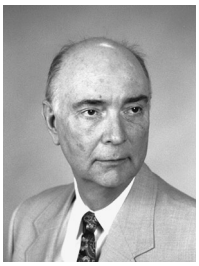
A Unified Mathematical Framework for Strapdown Algorithm Design

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

A unified two-speed mathematical framework is described for strapdown inertial system integration algorithm design that uses a new concept for velocity/position updating. The velocity/position equations are structured using a Jordan-like attitude updating approach; the update equations are designed to provide the exact solution under particular input conditions, and the update inputs are then redefined to provide the correct solution under general motion. For the Jordan approach, the attitude update input is the Euler rotation vector generated by high-speed integration of a rotation vector rate equation; for the new velocity/position updating concept, inputs are velocity/position translation vectors generated by high-speed integration of translation vector rate equations. Exact differential equations are derived for the translation vectors that parallel the exact rotation vector rate equation originally derived by Laning and applied by Bortz in a Jordan-like structure. The new velocity/position concept coupled with the Jordan/Bortz/Laning attitude updating approach provides a unified framework for strapdown integration algorithm design. Continuous-form algorithms are developed within the unified framework based on simplified forms of the exact rotation/translation vector rate equations. Algorithm performance comparisons are presented based on derived analytical error equations under maneuver and vibration motion. A discussion is included on algorithm design approaches for digital integration of the rotation/translation vector rate equations. Simulation studies are described that numerically validate the accuracy of the principal analytical results.

Nomenclature

$\boldsymbol{a}_{\mathrm{SF}}, a_{\mathrm{SF}_x}, a_{\mathrm{SF}_y}, a_{\mathrm{SF}_z}$	=	specific force acceleration vector (sensed by accelerometers) and components in frame B
a_{SF_0}	=	sinusoidal specific force acceleration vibration amplitude
a_η, a_ζ	=	coefficients in F, G inverses
$\boldsymbol{a}_0, \dot{\boldsymbol{a}}_0, \ddot{\boldsymbol{a}}_0, \ldots$	=	value and derivatives of $\boldsymbol{a}_{\mathrm{SF}}$ at t_{m-1}
B	=	sensor (body) coordinate frame aligned with strapdown inertial sensor axes
B_m	=	frame B orientation at computer cycle m, treated as an inertially nonrotating coordinate frame
$B(t)$	=	frame-B orientation at time t
b_η, b_ζ	=	coefficients in F, G inverses
C	=	shorthand for $C_{B(t)}^{B_{m-1}}$
$C_{[\,]}^{(\,)}$	=	direction cosine matrix that transforms vectors from coordinate frame [ ] to ( )
F	=	matrix for translating $\boldsymbol{\eta}$ into $\Delta\boldsymbol{v}_{\mathrm{SF}}$
$f_1, f_2, \ldots$	=	trigonometric coefficients in unified navigation equations
G	=	matrix for translating $\boldsymbol{\zeta}$ into $\Delta\boldsymbol{R}_{\mathrm{SF}}$
G_1, G_2	=	sinusoids at folded generalized vibration frequency ρ'
$\boldsymbol{g}$	=	gravitational acceleration
g_1, g_2, g_3, g_4	=	coefficient functions in algorithm errors under vibration
H	=	functional operator for $\dot{\boldsymbol{\chi}}$
h_1, h_2	=	coefficients in rotation/translation vector rate equations
I	=	identity matrix
J	=	function used in G_1, G_2 folded vibration frequency sinusoids
K	=	nearest integer number of generalized ρ frequency cycles in an m cycle
M	=	number of m cycle sums
m	=	computer cycle index; as subscript, indicates parameter value at computer cycle m
N	=	navigation coordinate frame (inertially nonrotating for this paper)
$\boldsymbol{R}$	=	position vector
$\boldsymbol{S}_\alpha$	=	double integral of $\boldsymbol{\omega}$ since computer cycle $m-1$



Paul G. Savage is president of Strapdown Associates, Inc. (www.strapdownassociates.com), a company he founded in 1980 that provides strapdown inertial system design, consulting, and educational services. Since 1981, Mr. Savage continues to provide his course Introduction to Strapdown Inertial Navigation Systems to the aerospace industry. He has recently written and published the text *Strapdown Analytics*. From 1963 to 1980 Mr. Savage led design teams at Honeywell in the evolutionary development of laser gyro strapdown inertial navigation systems. Mr. Savage received his M.S. and B.S. degrees in aeronautical engineering from Massachusetts Institute of Technology in 1960. He is a Senior Member of AIAA.

Received 11 April 2005; revision received 20 June 2005; accepted for publication 20 June 2005.

$\boldsymbol{S}_\upsilon$ = double integral of $\boldsymbol{a}_{\mathrm{SF}}$ since computer cycle $m-1$

T_m = time interval for m computer update cycle

t = time

t_0 = time t at $m=0$

$\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z$ = unit vectors along x, y, zB-frame axes

$\boldsymbol{V}, \boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3$ = general vector parameters

$\boldsymbol{v}$ = velocity vector

$\boldsymbol{\alpha}$ = integral of $\boldsymbol{\omega}$ in frame B since computer cycle $m-1$

$\alpha, \alpha_x, \alpha_y, \alpha_z$ = magnitude and B-frame components of $\boldsymbol{\alpha}$

β = vibration frequency parameter ΩT_m

ΔC = trigonometric function in algorithm errors under vibration

$\Delta \boldsymbol{R}_g$ = change in position due to gravity since computer cycle $m-1$

$\Delta \boldsymbol{R}_{\mathrm{SF}}$ = change in position due to specific force acceleration since computer cycle $m-1$

$\Delta \boldsymbol{R}_{\mathrm{SF_{Algo/d}}}$ = algorithm computed position change due to specific force acceleration when using $\Delta\boldsymbol{\zeta}_{\mathrm{Algo/d}}$ for scrolling

$\Delta R_{\mathrm{SF_{max}}}$ = maximum value of $\Delta \boldsymbol{R}_{\mathrm{SF}}$ over computer cycle m

ΔS = trigonometric function in algorithm errors under vibration

$\Delta \boldsymbol{v}_g$ = change in velocity due to gravity since computer cycle $m-1$

$\Delta \boldsymbol{v}_{\mathrm{SF}}$ = change in velocity due to specific force acceleration since computer cycle $m-1$

$\Delta v_{\mathrm{SF_{max}}}$ = maximum magnitude of $\Delta \boldsymbol{v}_{\mathrm{SF}}$ over computer cycle m

$\Delta\boldsymbol{\zeta}_{\mathrm{Algo/c}}$ = scrolling portion of $\boldsymbol{\zeta}_{\mathrm{Algo/c}}$

$\Delta\boldsymbol{\zeta}_{\mathrm{Algo/d}}$ = alternative definition for scrolling term

$\Delta\boldsymbol{\zeta}_i$ = ith $\boldsymbol{\zeta}$ Picard expansion solution component

$\Delta\boldsymbol{\eta}_{\mathrm{Algo/c}}$ = sculling portion of $\boldsymbol{\eta}_{\mathrm{Algo/c}}$

$\Delta\boldsymbol{\eta}_i$ = ith $\boldsymbol{\eta}$ Picard expansion solution component

$\Delta\boldsymbol{\phi}_i$ = ith $\boldsymbol{\phi}$ Picard expansion solution component

$\Delta\boldsymbol{\chi}_i$ = ith $\boldsymbol{\chi}$ Picard expansion solution component

$\delta\Delta\boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}}$ = computation error in exact solution for $\Delta\boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}}$

$\delta\Delta\boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}}$ = computation error in exact solution for $\Delta\boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}}$

$\delta(\delta\boldsymbol{\phi}_{\mathrm{AnalAlgcEr}})$, $\delta(\delta\boldsymbol{\eta}_{\mathrm{AnalAlgcEr}})$ = error in the analytical estimates for $\delta\boldsymbol{\phi}_{\mathrm{AlgcEr}}$, $\delta\boldsymbol{\eta}_{\mathrm{AlgcEr}}$

$\delta\boldsymbol{\zeta}_{\mathrm{Algo/a}}, \delta\boldsymbol{\zeta}_{\mathrm{Algo/b}}, \delta\boldsymbol{\zeta}_{\mathrm{Algo/c}}$ = errors in $\boldsymbol{\zeta}_{\mathrm{Algo/a}}, \boldsymbol{\zeta}_{\mathrm{Algo/b}}, \boldsymbol{\zeta}_{\mathrm{Algo/c}}$

$\delta\boldsymbol{\zeta}_{\mathrm{Pic}}$ = error in Picard expansion solution for $\boldsymbol{\zeta}$

$\delta\boldsymbol{\eta}_{\mathrm{AlgcEr}}, \delta\eta_{\mathrm{AlgcEr_{Mag}}}$ = error in $\boldsymbol{\eta}_{\mathrm{Algo/c}}$ and the error magnitude

$\delta\boldsymbol{\eta}_{\mathrm{Algo/a}}, \delta\boldsymbol{\eta}_{\mathrm{Algo/b}}, \delta\boldsymbol{\eta}_{\mathrm{Algo/c}}$ = errors in $\boldsymbol{\eta}_{\mathrm{Algo/a}}, \boldsymbol{\eta}_{\mathrm{Algo/b}}, \boldsymbol{\eta}_{\mathrm{Algo/c}}$

$\delta\boldsymbol{\eta}_{\mathrm{Pic}}$ = error in Picard expansion solution for $\boldsymbol{\eta}$

$\delta\boldsymbol{\phi}_{\mathrm{AlgcEr}}, \delta\boldsymbol{\phi}_{\mathrm{AlgcEr_{Mag}}}$ = error in $\boldsymbol{\phi}_{\mathrm{Algo/c}}$ and the error magnitude

$\delta\boldsymbol{\phi}_{\mathrm{Algo/a}}, \delta\boldsymbol{\phi}_{\mathrm{Algo/b}}, \delta\boldsymbol{\phi}_{\mathrm{Algo/c}}$ = errors in $\boldsymbol{\phi}_{\mathrm{Algo/a}}, \boldsymbol{\phi}_{\mathrm{Algo/b}}, \boldsymbol{\phi}_{\mathrm{Algo/c}}$

$\delta\boldsymbol{\phi}_m$ = computation error in exact solution for $\boldsymbol{\phi}_m$

$\delta\boldsymbol{\phi}_{\mathrm{Pic}}$ = error in Picard expansion solution for $\boldsymbol{\phi}$

$\boldsymbol{\zeta}$ = position translation vector in the B frame

$\boldsymbol{\zeta}_{\mathrm{Algo/a}}, \boldsymbol{\zeta}_{\mathrm{Algo/b}}, \boldsymbol{\zeta}_{\mathrm{Algo/c}}$ = simplified versions of $\boldsymbol{\zeta}$ for algorithm usage

$\boldsymbol{\zeta}_i$ = ith-order Picard expansion solution for $\boldsymbol{\zeta}$

ζ_{max} = maximum magnitude of $\boldsymbol{\zeta}$ over computer cycle m

$\boldsymbol{\zeta}_{\mathrm{Scroll}}$ = portion of $\boldsymbol{\zeta}$ due to scrolling motion

$\boldsymbol{\eta}$ = velocity translation vector in the B frame

$\boldsymbol{\eta}_{\mathrm{Algo/a}}, \boldsymbol{\eta}_{\mathrm{Algo/b}}, \boldsymbol{\eta}_{\mathrm{Algo/c}}$ = simplified versions of $\boldsymbol{\eta}$ for algorithm usage

$\boldsymbol{\eta}_i$ = ith-order Picard expansion solution for $\boldsymbol{\eta}$

η_{max} = maximum magnitude of $\boldsymbol{\eta}$ over computer cycle m

$\boldsymbol{\eta}_{\mathrm{Scull}}$ = portion of $\boldsymbol{\eta}$ due to sculling motion

θ_0 = sinusoidal angular vibration amplitude

ρ = generalized vibration frequency parameter

ρ' = folded frequency of sinusoids at frequency ρ sampled at the m cycle rate

τ = normalized time since computer cycle $m-1$ as a fraction of T_m

$\boldsymbol{\upsilon}, \upsilon_x, \upsilon_y, \upsilon_z$ = integral of $\boldsymbol{a}_{\mathrm{SF}}$ in frame B since computer cycle $m-1$ and its components

$\boldsymbol{\phi}, \phi$ = rotation vector equivalent to $C_{B(t)}^{B_{m-1}}$ and its magnitude

$\boldsymbol{\phi}_{\mathrm{Algo/a}}, \boldsymbol{\phi}_{\mathrm{Algo/b}}, \boldsymbol{\phi}_{\mathrm{Algo/c}}$ = simplified versions of $\boldsymbol{\phi}$ for algorithm usage

$\boldsymbol{\phi}_{\mathrm{cone}}$ = portion of $\boldsymbol{\phi}$ due to coning motion

$\boldsymbol{\phi}_i$ = ith-order Picard expansion solution for $\boldsymbol{\phi}$

ϕ_{max} = maximum magnitude of $\boldsymbol{\phi}$ over computer cycle m

$\boldsymbol{\chi}$ = state vector containing $\boldsymbol{\phi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$

$\boldsymbol{\chi}_i$ = ith-order Picard expansion solution for $\boldsymbol{\chi}$

Ω = vibration frequency

Ω' = folded frequency of sinusoids at frequency Ω sampled at the m cycle rate

$\boldsymbol{\omega}, \boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z$ = angular rate vector and components in the B frame

$\boldsymbol{\omega}_0, \dot{\boldsymbol{\omega}}_0, \ddot{\boldsymbol{\omega}}_0, \ldots$ = value and derivatives of $\boldsymbol{\omega}$ at t_{m-1}

$(\,)^{[\,]}$ = column matrix formed from the components of vector ( ) in coordinate frame [ ]

$[(\,)\times]$ = skew symmetric cross-product matrix form of vector ( ) that satisfies $[(\,)\times]\,\boldsymbol{V} = (\,)\times\boldsymbol{V}$

$\dot{(\,)}, \ddot{(\,)}, \dddot{(\,)}$ = first, second and third time derivatives of ( )

Introduction

MODERN day strapdown inertial navigation system integration routines have been formulated in many applications using a two-speed structure; navigation parameters (attitude, velocity, position) are updated using basic higher-order algorithms, and inputs to the basic algorithms are derived from a high-speed digital integration process within the navigation parameter update time interval. The basic algorithms account for low-frequency large angular amplitude dynamics; the high-frequency routines account for high-frequency small angular amplitude effects (e.g., vibration). The overall structure is designed such that most of the computations are

executed by the basic updating function with the high-speed operation containing comparatively few calculations per high-speed cycle. The net result is that computer through put is conserved because only the high-speed computations must be operated at the rate required to process high-frequency angular dynamics accurately; the basic navigation parameter update frequency can be set based on other less demanding requirements (e.g., to ensure that the maximum attitude change per update cycle is small, thereby protecting small angle approximations in the processing algorithms). When the basic algorithms (the dominant navigation solution) are formulated from closed-form analytical expressions that are exact under particular input conditions (e.g., constant strapdown angular rate and specific force acceleration) documentation is straightforward, validation is precise (by exact comparison with comparable exact solution truth models), and one set of algorithms can be used generically for all applications.

The two-speed structure for attitude updating was originated by the author in 1966 using a second-order basic updating algorithm with the high-speed digital integration function derived from a first-order differential equation.[1] In 1969, Jordan proposed a two-speed attitude computation structure in which the form of the basic updating operation was based on the correct attitude solution under rotation about a fixed (nonrotating) axis.[2] For this condition, the algorithm input is the direct integral of angular rate vector components provided by angular rate sensors. For general angular motion (in which the axis of rotation is changing direction), Jordan proposed that the analytical form of the basic algorithm should remain the same, but that the Euler rotation vector be used for input rather than integrated angular rate. The high-speed operation then became an integration of rotation vector rate-of-change over the attitude update cycle. The Jordan high-speed integration algorithm for the rotation vector was based on an approximate first order application of the Goodman/Robinson theorem.[3]

In 1971, Bortz[4] proposed that the high speed integration operation in the two-speed attitude updating structure be based on the exact rotation vector rate equation derived in 1949 by Laning.[5] Having the exact rotation vector rate equation as a base provided a framework for simplified algorithm development and accuracy evaluation by comparison with the exact form. The exact Jordan structure coupled with the integrated Bortz/Laning exact rotation vector rate equation has been the basis for continuing modern day strapdown attitude algorithm development.

This paper presents a new concept for strapdown velocity and position updating using a Jordan two-speed attitude computation structure. In direct analogy to Jordan attitude updating, the analytical form of the velocity/position algorithms is based on solutions that are exact under particular input conditions, in this case constant angular rate and specific force acceleration (Ref. 6 and Ref. 7, Secs. 7.2.2.2.1 and 7.3.3.1). For constant angular-rate/specific-force, the algorithm input is the direct integral of angular-rate/specific-force vector components provided by angular-rate sensors and accelerometers. The exact constant-input solutions were originally derived as an expansion of rotation compensation terms present in first-order two-speed algorithms designed for general (nonconstant) inputs (Refs. 6, 8, and Ref. 7, Secs. 7.2.2.2 and 7.3.3). For the new concept, exact general-input velocity/position algorithms are synthesized using the Jordan approach of having the same analytical form under general input as the exact constant-input solutions, but with velocity/position translation vectors (analogous to the rotation vector) replacing the integrated angular-rate/specific-force used under constant input. The translation vectors are calculated by integrating translation vector rate equations over a velocity/position update cycle. Differential equations are derived in the paper for the velocity/position translation vectors that are exact under general motion (analogous to the Laning rotation vector rate equation). The new velocity/position updating concept coupled with the Jordan/Bortz/Laning attitude updating approach provides a unified mathematical framework for strapdown integration algorithm design.

Strapdown computation algorithms can be designed within the unified framework using the exact closed-form equations directly (without approximation) for attitude/velocity/position updating. Inputs to the updating algorithms would be high-speed numerical integration routines based on simplified integral versions of the exact rotation/translation vector rate equations. The paper provides examples of simplified rotation/translation vector rate equations and associated performance characteristics under generalized angular-rate/specific-force maneuver and vibration profiles. The performance investigations and some of the high speed routines are based on a Picard expansion solution[9] to the exact rotation/translation vector rate equations, derived here in powers of integrated angular-rate/specific-force.

A general discussion is included in the paper on potential design approaches for digital algorithms used for high-speed integration of the simplified rotation/translation vector rate equations over the attitude/velocity/position update cycle. The Appendix provides a description of digital simulation studies conducted to numerically verify the accuracy of the principal analytical results.

Unified Mathematical Framework Based on Constant Angular Rate/Specific Force

The following generalized differential equations describe time rates of change of attitude and specific-force acceleration induced velocity/position in a coordinate frame B_{m-1} representing the orientation of a generalized rotating coordinate frame B at time t_{m-1} (Ref. 7, Secs. 4.1, 4.3, and 4.4.1.2):

$$\dot{C}_{B(t)}^{B_{m-1}} = C_{B(t)}^{B_{m-1}}(\boldsymbol{\omega}\times) \qquad \Delta\dot{\boldsymbol{v}}_{\mathrm{SF}}^{B_{m-1}}(t) = C_{B(t)}^{B_{m-1}}\boldsymbol{a}_{\mathrm{SF}}$$

$$\Delta\dot{\boldsymbol{R}}_{\mathrm{SF}}^{B_{m-1}}(t) = \Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{(m-1)}}(t) \tag{1}$$

A generalized structure for updating attitude/velocity/position in a strapdown inertial navigation system is obtained from the cumulative integral of Eqs. (1) over strapdown computer update cycles m, referenced to a nonrotating coordinate frame N and including the effect of gravity,

$$C_{B(t)}^{B_{m-1}} = I + \int_{t_{m-1}}^{t} \dot{C}_{B(t)}^{B_{m-1}}\,\mathrm{d}t \qquad C_{B_m}^{N} = C_{B_{m-1}}^{N} C_{B(t_m)}^{B_{m-1}}$$

$$\boldsymbol{v}_m^N = \boldsymbol{v}_{m-1}^N + \Delta\boldsymbol{v}_g^N(t_m) + C_{B_{m-1}}^N \Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t_m)$$

$$\Delta\boldsymbol{v}_g^N(t) = \int_{t_{m-1}}^{t} \boldsymbol{g}^N\,\mathrm{d}t \qquad \Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t) = \int_{t_{m-1}}^{t} \Delta\dot{\boldsymbol{v}}_{\mathrm{SF}}^{B_{m-1}}\,\mathrm{d}t$$

$$\boldsymbol{R}_m^N = \boldsymbol{R}_{m-1}^N + \boldsymbol{v}_{m-1}^N T_m + \Delta\boldsymbol{R}_g^N(t_m) + C_{B_{m-1}}^N \Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t_m)$$

$$\Delta\boldsymbol{R}_g^N(t) = \int_{t_{m-1}}^{t} \Delta\boldsymbol{v}_g^N(t)\,\mathrm{d}t \qquad \Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t) = \int_{t_{m-1}}^{t} \Delta\dot{\boldsymbol{R}}_{\mathrm{SF}}^{B_{m-1}}(t)\,\mathrm{d}t \tag{2}$$

In a strapdown inertial navigation system, the angular rate vector $\boldsymbol{\omega}$ is measured by strapdown angular rate sensors, the specific force acceleration vector $\boldsymbol{a}_{\mathrm{SF}}$ is measured by strapdown accelerometers, the B frame represents a coordinate frame that maintains alignment with the rotating strapdown sensors (the body frame), and the N frame represents navigation coordinates for output reporting. In many systems, the N frame is slowly rotated (e.g., to maintain one axis vertical in the presence of vehicle motion and Earth's rotation rate) in which case Eqs. (2) would have additional terms (Refs. 6, 8, 10, and Ref. 7, Secs. 4.1, 4.3, and 4.4.1.2). Note that in some systems the C_B^N direction cosine matrix is replaced by an equivalent attitude quaternion (using the quaternion equivalent to $C_{B(t_m)}^{B_{m-1}}$ in Eqs. (2) for input), and position location $\boldsymbol{R}$ is represented by altitude and angular location over the Earth's surface [using the $C_{B_{m-1}}^N \Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t_m)$ term in Eqs. (2) for input (Refs. 6, 8, 10, and Ref. 7, Secs. 4.1, 4.4, 7.1.2, and 7.3)]. Equations (2) and subsequent results in this paper can be easily modified to incorporate these alternative navigation parameter representations.

The unified approach to navigation parameter updating is based on the following closed-form solution to the Eqs. (2) attitude and specific-force acceleration integrals under conditions when $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ are constant (Ref. 6 and Ref. 7, Secs. 7.2.2.2 and 7.3.3):

$$C_{B(t)}^{B_{m-1}} = I + f_1(\alpha)(\boldsymbol{\alpha}\times) + f_2(\alpha)(\boldsymbol{\alpha}\times)^2 \tag{3}$$

$$\Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t) = \left[I + f_2(\alpha)(\boldsymbol{\alpha}\times) + f_3(\alpha)(\boldsymbol{\alpha}\times)^2\right]\boldsymbol{\upsilon} \tag{4}$$

$$\Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t) = \left[I + 2f_3(\alpha)(\boldsymbol{\alpha}\times) + 2f_4(\alpha)(\boldsymbol{\alpha}\times)^2\right]\boldsymbol{S}_{\upsilon} \tag{5}$$

$$\boldsymbol{\alpha} \equiv \int_{t_{m-1}}^{t} \boldsymbol{\omega}\,\mathrm{d}t \qquad \boldsymbol{\upsilon} \equiv \int_{t_{m-1}}^{t} \boldsymbol{a}_{\mathrm{SF}}\,\mathrm{d}t \qquad \boldsymbol{S}_{\upsilon} \equiv \int_{t_{m-1}}^{t} \boldsymbol{\upsilon}\,\mathrm{d}t \tag{6}$$

with

$$f_1(\alpha) = \sin\alpha/\alpha \qquad f_2(\alpha) = (1-\cos\alpha)/\alpha^2$$
$$f_3(\alpha) = (1/\alpha^2)(1-f_1(\alpha)) \qquad f_4(\alpha) = (1/\alpha^2)\left(\tfrac{1}{2} - f_2(\alpha)\right) \tag{7}$$

Note that Eqs. (3) and (4) are actually valid in a more general environment; Eq. (3) is valid when the direction of $\boldsymbol{\omega}$ is fixed even though its magnitude may be changing, and Eq. (4) is valid under the previous $\boldsymbol{\omega}$ condition and when the ratio of $\boldsymbol{a}_{\mathrm{SF}}$ to $\boldsymbol{\omega}$ components is constant. Constant angular-rate/specific-force is a special case of the preceding more general conditions.

Unified Mathematical Framework Based on General Motion

For the unified updating approach, under general motion conditions the form of Eqs. (3–7) is maintained, but $\boldsymbol{\alpha}$, $\boldsymbol{\upsilon}$, and $\boldsymbol{S}_{\upsilon}$ are replaced by the rotation and velocity/position translation vectors $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$,

$$C_{B(t)}^{B_{m-1}} = I + f_1(\phi)(\boldsymbol{\phi}\times) + f_2(\phi)(\boldsymbol{\phi}\times)^2$$
$$\Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t) = \left[I + f_2(\phi)(\boldsymbol{\phi}\times) + f_3(\phi)(\boldsymbol{\phi}\times)^2\right]\boldsymbol{\eta}$$
$$\Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t) = \left[I + 2f_3(\phi)(\boldsymbol{\phi}\times) + 2f_4(\phi)(\boldsymbol{\phi}\times)^2\right]\boldsymbol{\zeta}$$
$$f_1(\phi) = \sin\phi/\phi = 1 - \phi^2/3! + \phi^4/5! - \cdots$$
$$f_2(\phi) = (1-\cos\phi)/\phi^2 = 1/2! - \phi^2/4! + \phi^4/6! - \cdots$$
$$f_3(\phi) = (1/\phi^2)(1-f_1(\phi)) = 1/3! - \phi^2/5! + \cdots$$
$$f_4(\phi) = (1/\phi^2)\left(\tfrac{1}{2} - f_2(\phi)\right) = 1/4! - \phi^2/6! + \cdots \tag{8}$$

The rotation vector $\boldsymbol{\phi}$ in Eqs. (8) is as described by Laning[5] and Jordan.[2] The translation vectors $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ are new concepts. They are defined implicitly to be vectors that when used as shown in Eqs. (8) yield the correct result for $\Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t)$ and $\Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t)$ under general motion. This is also how the rotation vector $\boldsymbol{\phi}$ can be defined, that is, the vector that when used as shown in Eqs. (8) yields the correct result for $C_{B(t)}^{B_{m-1}}$. The definition of the correct result is that obtained by direct integration of generalized Eqs. (1).

Differential Equations for Translation and Rotation Vectors

The definitions for the velocity/position translation vectors given in the preceding section can be restated as $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ being vectors that equate the derivative of $\Delta\boldsymbol{v}_{\mathrm{SF}}^{B_{m-1}}(t)$, $\Delta\boldsymbol{R}_{\mathrm{SF}}^{B_{m-1}}(t)$ in Eqs. (8) to the $\Delta\dot{\boldsymbol{v}}_{\mathrm{SF}}^{B_{m-1}}(t)$, $\Delta\dot{\boldsymbol{R}}_{\mathrm{SF}}^{B_{m-1}}(t)$ terms in Eqs. (1). Thus, $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ can be defined as the solutions to

$$C\boldsymbol{a}_{\mathrm{SF}} = \dot{F}\boldsymbol{\eta} + F\dot{\boldsymbol{\eta}} \qquad F\boldsymbol{\eta} = \dot{G}\boldsymbol{\zeta} + G\dot{\boldsymbol{\zeta}} \tag{9}$$
$$C = I + f_1(\phi\times) + f_2(\phi\times)^2$$
$$F = I + f_2(\phi\times) + f_3(\phi\times)^2$$
$$G = I + 2f_3(\phi\times) + 2f_4(\phi\times)^2 \tag{10}$$

where f_1–f_4 are as defined in Eqs. (8). The $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$ rate equations, that is, $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$, are derived directly from Eqs. (9) and (10). Solving for $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$ in Eq. (9) gives

$$\dot{\boldsymbol{\eta}} = F^{-1}(C\,\boldsymbol{a}_{\mathrm{SF}} - \dot{F}\boldsymbol{\eta}) \qquad \dot{\boldsymbol{\zeta}} = G^{-1}(F\boldsymbol{\eta} - \dot{G}\boldsymbol{\zeta}) \tag{11}$$

Analytical expressions for the F and G inverses in Eqs. (11) can be developed by assuming a general form for F^{-1} and G^{-1} of $I - a(\phi\times) + b(\phi\times)^2$, multiplying F and G by the general form, equating the products to the identity matrix, substituting the identity $(\phi\times)^3 = -\phi^2(\phi\times)$, and solving for the a and b coefficients. The result is

$$F^{-1} = I - a_\eta(\phi\times) + b_\eta(\phi\times)^2$$
$$G^{-1} = I - a_\zeta(\phi\times) + b_\zeta(\phi\times)^2$$
$$a_\eta = \frac{1}{2} \qquad b_\eta = \frac{1}{\phi^2}\left(1 - \frac{\phi\sin\phi}{2(1-\cos\phi)}\right)$$
$$a_\zeta = \frac{f_3}{2\left(f_2^2 + f_3^2\phi^2\right)} \qquad b_\zeta = \frac{f_3^2 - f_2 f_4}{f_2^2 + f_3^2\phi^2} \tag{12}$$

The derivative terms in Eqs. (11) are

$$\dot{F} = \dot{f}_2(\phi\times) + \dot{f}_3(\phi\times)^2 + f_2(\dot{\phi}\times) + f_3[(\phi\times)(\dot{\phi}\times) + (\phi\times)(\dot{\phi}\times)]$$
$$\dot{G} = 2\dot{f}_3(\phi\times) + 2\dot{f}_4(\phi\times)^2 + 2f_3(\dot{\phi}\times) + 2f_4[(\phi\times)(\dot{\phi}\times) + (\phi\times)(\dot{\phi}\times)] \tag{13}$$

When the f_i definitions in Eqs. (8) are used with $\dot{f}_i = (\mathrm{d}f_i/\mathrm{d}\phi)\dot{\phi}$ and $\dot{\phi} = (1/\phi)\boldsymbol{\phi}\cdot\boldsymbol{\omega}$, (from Refs. 4, 5, 8, and Ref. 7, Sec. 3.3.5) the $\dot{f}_i$ terms in Eqs. (13) become

$$\dot{f}_2 = \frac{f_1 - 2f_2}{\phi^2}\boldsymbol{\phi}\cdot\boldsymbol{\omega} \qquad \dot{f}_3 = \frac{f_2\phi^2 - 3(1-f_1)}{\phi^4}\boldsymbol{\phi}\cdot\boldsymbol{\omega}$$
$$\dot{f}_4 = -\frac{1 + f_1 - 4f_2}{\phi^4}\boldsymbol{\phi}\cdot\boldsymbol{\omega} \tag{14}$$

Substituting Eqs. (10), (12), and (13) with Eqs. (14) into Eqs. (11) yields expanded expressions for $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$. The final $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$ form shown subsequently in Eqs. (15) is obtained after a significant amount of matrix/vector algebraic expansion/manipulation, trigonometric manipulation/compression of coefficients, application of $\phi\dot{\phi} = \boldsymbol{\phi}\cdot\boldsymbol{\omega}$, and use of the identities,

$$\boldsymbol{V}_1 \times (\boldsymbol{V}_2 \times \boldsymbol{V}_3) = \boldsymbol{V}_2(\boldsymbol{V}_1\cdot\boldsymbol{V}_3) - \boldsymbol{V}_3(\boldsymbol{V}_1\cdot\boldsymbol{V}_2)$$
$$(\boldsymbol{\phi}\times)^3 = -\phi^2(\boldsymbol{\phi}\times)$$
$$[(\boldsymbol{V}_1 \times \boldsymbol{V}_2)\times] = (\boldsymbol{V}_1\times)(\boldsymbol{V}_2\times) - (\boldsymbol{V}_2\times)(\boldsymbol{V}_1\times)$$

The rotation vector rate equation $\dot{\boldsymbol{\phi}}$ is also included in Eqs. (15) for comparison with $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$. Several derivations for $\dot{\boldsymbol{\phi}}$ are available in the literature (e.g., Refs. 4, 8, 10, 11 and Ref. 7, Sec. 3.3.5) in addition to Laning's original derivation.[5] Thus,

$$\begin{aligned}
\dot{\boldsymbol{\phi}} &= \boldsymbol{\omega} + \tfrac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega} + f_5\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\omega})\\
\dot{\boldsymbol{\eta}} &= \boldsymbol{a}_{\mathrm{SF}} + \tfrac{1}{2}(\boldsymbol{\phi}\times\boldsymbol{a}_{\mathrm{SF}} - \dot{\boldsymbol{\phi}}\times\boldsymbol{\eta})\\
&\quad + f_5\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{a}_{\mathrm{SF}} - \dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}) + f_3(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\eta}\\
&\quad - \tfrac{1}{2}f_3\boldsymbol{\phi}\times[(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\eta}] + f_4\phi^2\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta} + f_6\boldsymbol{\phi}\times[\boldsymbol{\phi}\times(\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta})]\\
&\quad + \tfrac{1}{2}f_3\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times\boldsymbol{\eta} + f_7\boldsymbol{\phi}\times\{\boldsymbol{\phi}\times[(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\eta}]\}\\
&\quad + f_8\phi^2\boldsymbol{\phi}\times(\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}) - f_8\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\eta})\\
\dot{\boldsymbol{\zeta}} &= \boldsymbol{\eta} + \tfrac{1}{6}(\boldsymbol{\phi}\times\boldsymbol{\eta} - 2\dot{\boldsymbol{\phi}}\times\boldsymbol{\zeta})\\
&\quad + f_9\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\eta} - 2\dot{\boldsymbol{\phi}}\times\boldsymbol{\zeta}) + 2f_4(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\zeta}\\
&\quad - f_{10}\boldsymbol{\phi}\times[(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\zeta}] - f_{11}\phi^2\boldsymbol{\phi}\times\boldsymbol{\eta} + f_{12}\phi^2\dot{\boldsymbol{\phi}}\times\boldsymbol{\zeta}\\
&\quad + f_{13}\boldsymbol{\phi}\times[\boldsymbol{\phi}\times(\dot{\boldsymbol{\phi}}\times\boldsymbol{\zeta})] + f_{14}\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times\boldsymbol{\zeta} + f_{15}\boldsymbol{\phi}\times\{\boldsymbol{\phi}\\
&\quad \times[(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}})\times\boldsymbol{\zeta}]\} + f_{16}\phi^2\boldsymbol{\phi}\times(\dot{\boldsymbol{\phi}}\times\boldsymbol{\zeta}) - f_{17}\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\zeta})
\end{aligned} \tag{15}$$

In Eqs. (15), f_3 and f_4 are as in Eqs. (8) and h_1, h_2, f_5–f_{17} are given by

$$f_5 = \frac{1}{\phi^2}\left[1 - \frac{\phi \sin\phi}{2(1-\cos\phi)}\right] \qquad f_6 = \frac{1}{\phi^2}\left(1 - \frac{1}{2}f_1 - f_2\right)$$

$$f_7 = f_3 f_5 \qquad f_8 = \frac{1}{2\phi^4}\left[5 + \cos\phi - \frac{3\phi\sin\phi}{(1-\cos\phi)}\right]$$

$$f_9 = f_3 - f_2 h_1 + h_2\left(1 - f_3\phi^2\right)$$

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

$$f_{12} = \frac{2}{\phi^2}\left(\frac{1}{6} - f_3\right) \qquad f_{13} = 2(2 f_4 h_1 - f_3 h_2)$$

$$f_{14} = \frac{2}{\phi^4}\left\{(1 + f_1 - 4f_2)h_1\phi^2 - \left[f_2\phi^2 - 3(1-f_1)\right]\left(1 - h_2\phi^2\right)\right\}$$

$$f_{15} = 2 f_4 h_2$$

$$f_{16} = \frac{2}{\phi^2}\left\{f_3 - 2f_4 - h_1(f_2 - f_3) + h_2\left[1 - \left(f_3 - 2f_4\right)\phi^2\right]\right\}$$

$$f_{17} = -\frac{2}{\phi^4}\left\{\left[f_2\phi^2 - 3(1-f_1)\right]h_1 + (1 + f_1 - 4f_2)\left(1 - h_2\phi^2\right)\right\}$$

$$h_1 = \frac{f_3}{2\left(f_2^2 + f_3^2\phi^2\right)} \qquad h_2 = \frac{f_3^2 - f_2 f_4}{f_2^2 + f_3^2\phi^2} \tag{16}$$

Note the similarity in structure in Eqs. (15) for the $\dot{\boldsymbol{\phi}}$, $\dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$ equations out to second order in products of $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ (i.e., the first line of $\dot{\boldsymbol{\phi}}$ compared with the first two lines of $\dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$). The leading inputs to $\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\zeta}}$ are specific force acceleration $\boldsymbol{a}_{\mathrm{SF}}$ and its integral $\boldsymbol{\eta}$; for $\dot{\boldsymbol{\phi}}$, the leading input is angular rate $\boldsymbol{\omega}$. It can be verified analytically that under constant $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$, Eqs. (15) reduce exactly to $\dot{\boldsymbol{\phi}} = \boldsymbol{\omega}$, $\dot{\boldsymbol{\eta}} = \boldsymbol{a}_{\mathrm{SF}}$, and $\dot{\boldsymbol{\zeta}} = \boldsymbol{\upsilon}$, that is, $\boldsymbol{\phi} = \boldsymbol{\alpha}$, $\boldsymbol{\eta} = \boldsymbol{\upsilon}$, and $\boldsymbol{\zeta} = \boldsymbol{S}_{\upsilon}$. Hence, Eqs. (8) reduce exactly to Eqs. (3–7). This is the expected result because Eqs. (8) were formulated exactly from the Eqs. (3–7) constant angular-rate/specific-force solutions. As noted following Eqs. (8), it is also true that $\dot{\boldsymbol{\phi}}$ and $\dot{\boldsymbol{\eta}}$ reduce to $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ under the more general condition of constant $\boldsymbol{\omega}$ direction and constant ratio of $\boldsymbol{a}_{\mathrm{SF}}$ to $\boldsymbol{\omega}$ components (of which constant $\boldsymbol{\omega}$, $\boldsymbol{a}_{\mathrm{SF}}$ is a particular case); $\dot{\boldsymbol{\zeta}}$ reduces to $\boldsymbol{\upsilon}$ only under constant $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$. Thus, other than the leading $\boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}}$, and $\boldsymbol{\upsilon}$ inputs to $\dot{\boldsymbol{\phi}}, \dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$ in Eqs. (15), the additional terms (also known as coning for $\dot{\boldsymbol{\phi}}$, sculling for $\dot{\boldsymbol{\eta}}$, and scrolling for $\dot{\boldsymbol{\zeta}}$) measure contributions to $\dot{\boldsymbol{\phi}}, \dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$ caused by departures from the earlier defined particular angular-rate/specific-force conditions. In general, coning, sculling, and scrolling effects are small compared to $\boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}}$, and $\boldsymbol{\upsilon}$.

The integral of Eqs. (15) over an m cycle provides $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ for Eqs. (8) which, with Eqs. (2), provides the unified structure for attitude, velocity, and position updating,

$$C_{B_m}^N = C_{B_{m-1}}^N C_{B_m}^{B_{m-1}}$$

$$\boldsymbol{v}_m^N = \boldsymbol{v}_{m-1}^N + \Delta\boldsymbol{v}_{g_m}^N + C_{B_{m-1}}^N \Delta\boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}}$$

$$\boldsymbol{R}_m^N = \boldsymbol{R}_{m-1}^N + \boldsymbol{v}_{m-1}^N T_m + \Delta\boldsymbol{R}_{g_m}^N + C_{B_{m-1}}^N \Delta\boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}}$$

$$C_{B_m}^{B_{m-1}} = I + f_1(\boldsymbol{\phi}_m\times) + f_2(\boldsymbol{\phi}_m\times)^2$$

$$\Delta\boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}} = \left[I + f_2(\boldsymbol{\phi}_m\times) + f_3(\boldsymbol{\phi}_m\times)^2\right]\boldsymbol{\eta}_m$$

$$\Delta\boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}} = \left[I + 2f_3(\boldsymbol{\phi}_m\times) + 2f_4(\boldsymbol{\phi}_m\times)^2\right]\boldsymbol{\zeta}_m$$

$$\boldsymbol{\phi}_m = \int_{t_{m-1}}^{t_m} \dot{\boldsymbol{\phi}}\,\mathrm{d}t \qquad \boldsymbol{\eta}_m = \int_{t_{m-1}}^{t_m} \dot{\boldsymbol{\eta}}\,\mathrm{d}t \qquad \boldsymbol{\zeta}_m = \int_{t_{m-1}}^{t_m} \dot{\boldsymbol{\zeta}}\,\mathrm{d}t \tag{17}$$

Equations (15–17) constitute a complete set for updating the attitude matrix, the velocity, and the position vectors. The equations contain no approximations and are exact under general motion.

Strapdown Algorithm Design Based on Unified Mathematical Formulation

Strapdown system digital integration algorithms can be designed within the unified framework by using exact equations (17) as shown, that is, without approximation, for the attitude/velocity/position updating function, with high-speed digital integration algorithms designed for the rotation/translation vector inputs. The similarity in structure between the Eqs. (17) attitude, velocity, and position update expressions simplifies conversion into equivalent software instructions. Approximate forms of Eqs. (15) and (16) would be used as the design base for the rotation/translation vector digital integration algorithms. Examples of the latter operation are provided subsequently in the paper.

Structuring the algorithms such that they are primarily based on the Eqs. (17) exact closed-form solutions significantly simplifies the algorithm software validation process. Validation generally consists of operating the algorithms with simulated sensor inputs designed to exercise all algorithm elements, and then comparing algorithm outputs with equivalent data generated from exact truth model dynamic simulators, for example, see Ref. 7, Sec. 11.2. For properly derived and programmed algorithms, the comparison will yield identically zero difference, thereby providing a clear unambiguous algorithm software validation. Once validated, such algorithms can be used as a generic set suitable for all strapdown inertial applications. Associated algorithm documentation is also simplified because algorithm derivations are classical analytical formulations and explanations/numerical-error-analysis justification for application dependent approximations are not required (because there are none). Modern day strapdown system computer technology (high throughput, long word length, floating-point architecture) allows the general use of such exact solution algorithms without penalty.

The dominant portion of the rotation/translation vectors is the simple integral of angular-rate/specific-force signals. The additional coning, sculling, and scrolling terms are much smaller in magnitude. Consequently, associated algorithm design and validation approaches can vary by designer while still achieving reasonable performance results. Validation of the rotation/translation vector digital integration algorithms is achieved in two steps: 1) verification that Eqs. (15) approximations yield expected and acceptable errors under sample dynamic inputs and 2) verification that under prescribed inputs digital integration algorithms based on the approximations yield the same numerical results as the equivalent continuous integration process. Step 1 is facilitated by using the continuous integration of Eqs. (15) and (16) (or its Picard expansion series equivalent[9]) as an exact reference solution. A new Picard series expansion approach is described next for Eqs. (15) and (16) in powers of angular-rate/specific-force and their integrals.

Equivalent Differential Equations Based on Picard Series Expansion

Equations (15) are functions of the $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ parameters being calculated as well as measured inputs $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$. For accuracy studies and algorithm development, it is convenient to generate the equivalent version in terms of $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ (and their integrals) only. In strapdown applications, the m cycle update time period is selected to be short enough that, under maximum expected angular rates, $\boldsymbol{\phi}$ will be small, thereby assuring that the $\boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}}$, and $\boldsymbol{\eta}$ leading terms in Eqs. (15) will be dominant. This allows a new Picard iterative approach to be used in generating a power series equivalent to Eqs. (15) in terms only of $\boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}}$ and their integrals. The Picard approach applied to Eqs. (15) can be outlined by first defining the state vector,

$$\boldsymbol{\chi} \equiv (\boldsymbol{\phi}^T, \boldsymbol{\eta}^T, \boldsymbol{\zeta}^T)^T \tag{18}$$

We then define the ith Picard expansion solution as

$$\boldsymbol{\chi}_i \equiv \left(\boldsymbol{\phi}_i^T, \boldsymbol{\eta}_i^T, \boldsymbol{\zeta}_i^T\right)^T \tag{19}$$

in which i is the Picard expansion series number and the corresponding order of accuracy in the Picard expansion solution for $\boldsymbol{\chi}$ and its components. The order of accuracy is defined as the maximum number of products of integrated $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ input parameters $\boldsymbol{\alpha}$, $\boldsymbol{\upsilon}$, and $\boldsymbol{S}_\upsilon$ [defined in Eq. (6)] appearing in the $\boldsymbol{\chi}_i$ solution. When defining the accuracy of ordering for this section, $\boldsymbol{\alpha}$ and $\boldsymbol{\upsilon}$ are each of order one, whereas $\boldsymbol{S}_\upsilon$ is of order two. In general, the integral of a parameter increases its ordering by one, for example, $\boldsymbol{S}_\upsilon$ is the integral of first-order parameter $\boldsymbol{\upsilon}$, hence, is of order two and $\boldsymbol{\alpha}$ and $\boldsymbol{\upsilon}$ are the integrals of zero-order parameters $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$, hence, are of order one. Conversely, the derivative of a parameter will decrease its ordering by one. We then define the general Picard expansion equivalent to Eqs. (15) as

$$\dot{\boldsymbol{\chi}}_i = \sum_1^i \Delta\dot{\boldsymbol{\chi}}_j \qquad \Delta\boldsymbol{\chi}_j = \int_{t_{m-1}}^{t} \Delta\dot{\boldsymbol{\chi}}_j \,\mathrm{d}t \qquad \boldsymbol{\chi}_i = \sum_1^i \Delta\boldsymbol{\chi}_j \tag{20}$$

where $\Delta\boldsymbol{\chi}_i$ is the change in $\boldsymbol{\chi}_i$ from the $i-1$ value. Each $\Delta\dot{\boldsymbol{\chi}}_i$ is given by

$$\Delta\dot{\boldsymbol{\chi}}_i = \dot{\boldsymbol{\chi}}_i - \dot{\boldsymbol{\chi}}_{i-1} \tag{21}$$

which is the recursive derivative form of the middle term in Eqs. (20). The $\dot{\boldsymbol{\chi}}_i$ term in Eq. (21) is calculated based on Eqs. (15), which can be represented as

$$\dot{\boldsymbol{\chi}} = H(\boldsymbol{\chi}, \boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}}) \tag{22}$$

Using the previous ordering rule for derivatives, $\dot{\boldsymbol{\chi}}_i$ is calculated from Eq. (22) with $\boldsymbol{\chi}_{i-1}$ used for $\boldsymbol{\chi}$ in H and with H only including terms up to order $i-1$ in $\boldsymbol{\alpha}$, $\boldsymbol{\upsilon}$, and $\boldsymbol{S}_\upsilon$ products,

$$\dot{\boldsymbol{\chi}}_i = [H(\boldsymbol{\chi}_{i-1}, \boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}})]_{\text{truncated at } i-1} \tag{23}$$

Hence, Eq. (21) is equivalently

$$\Delta\dot{\boldsymbol{\chi}}_i = [H(\boldsymbol{\chi}_{i-1}, \boldsymbol{\omega}, \boldsymbol{a}_{\mathrm{SF}})]_{\text{truncated at } i-1} - \dot{\boldsymbol{\chi}}_{i-1} \tag{24}$$

The Picard solution components to fourth order for $\Delta\boldsymbol{\phi}_i$ and $\Delta\boldsymbol{\eta}_i$ and to fifth order for $\Delta\boldsymbol{\zeta}_i$ are obtained by iterating on Eqs. (24) and (20) using Eqs. (15) with Eqs. (16) for H. For compatibility with $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$ fourth-order and $\boldsymbol{\zeta}$ fifth-order Picard expansion accuracy, the following first-order truncated Taylor series $\boldsymbol{\phi}$ expansions (i.e., in error by $\boldsymbol{\phi}^2$ and higher powers) are used for the Eqs. (16) f coefficients appearing in the Picard expansion components:

$$\begin{gathered} f_3 \approx 1/6 \qquad f_4 \approx 1/24 \qquad f_5 \approx 1/12 \qquad f_6 \approx 1/8 \\ f_9 \approx 1/36 \qquad f_{10} \approx 1/36 \qquad f_{11} \approx 1/540 \\ f_{12} \approx 1/60 \qquad f_{13} \approx 5/108 \qquad f_{14} \approx 1/30 \end{gathered} \tag{25}$$

The initial values for $\boldsymbol{\chi}_{i-1}$ and $\dot{\boldsymbol{\chi}}_{i-1}$ in Eq. (24) (i.e., the $i=1$ values) are zero ($\boldsymbol{\chi}_{i-1} = \boldsymbol{\chi}_0 = 0$ and $\dot{\boldsymbol{\chi}}_0 = 0$) to start the iteration process.

When the described procedure is followed, Picard component solutions to Eqs. (15) and (16) are obtained as

$$\begin{aligned} &\Delta\dot{\boldsymbol{\phi}}_1 = \boldsymbol{\omega} \qquad \Delta\boldsymbol{\phi}_1 = \boldsymbol{\alpha} \qquad \Delta\dot{\boldsymbol{\phi}}_2 = (1/2)\boldsymbol{\alpha} \times \boldsymbol{\omega} \\ &\Delta\dot{\boldsymbol{\phi}}_3 = (1/2)\Delta\boldsymbol{\phi}_2 \times \boldsymbol{\omega} + (1/6)\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\phi}}_2 \\ &\Delta\dot{\boldsymbol{\phi}}_4 = (1/2)\left(\Delta\boldsymbol{\phi}_3 \times \boldsymbol{\omega} + (1/3)\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\phi}}_3\right) \\ &\qquad + (1/6)\Delta\boldsymbol{\phi}_2 \times \Delta\dot{\boldsymbol{\phi}}_2 + (1/36)\alpha^2\Delta\dot{\boldsymbol{\phi}}_2 \end{aligned} \tag{26}$$

$$\begin{aligned} &\Delta\dot{\boldsymbol{\eta}}_1 = \boldsymbol{a}_{\mathrm{SF}} \qquad \Delta\boldsymbol{\eta}_1 = \boldsymbol{\upsilon} \qquad \Delta\dot{\boldsymbol{\eta}}_2 = (1/2)(\boldsymbol{\alpha} \times \boldsymbol{a}_{\mathrm{SF}} - \boldsymbol{\omega} \times \boldsymbol{\upsilon}) \\ &\Delta\dot{\boldsymbol{\eta}}_3 = (1/2)(\Delta\boldsymbol{\phi}_2 \times \boldsymbol{a}_{\mathrm{SF}} - \boldsymbol{\omega} \times \Delta\boldsymbol{\eta}_2) \\ &\qquad + (1/6)(\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\eta}}_2 - \Delta\dot{\boldsymbol{\phi}}_2 \times \boldsymbol{\upsilon}) \\ &\Delta\dot{\boldsymbol{\eta}}_4 = (1/2)[\Delta\boldsymbol{\phi}_3 \times \boldsymbol{a}_{\mathrm{SF}} - \boldsymbol{\omega} \times \Delta\boldsymbol{\eta}_3 + (1/3)(\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\eta}}_3 \\ &\qquad - \Delta\dot{\boldsymbol{\phi}}_3 \times \boldsymbol{\upsilon})] + (1/6)(\Delta\boldsymbol{\phi}_2 \times \Delta\dot{\boldsymbol{\eta}}_2 - \Delta\dot{\boldsymbol{\phi}}_2 \times \Delta\boldsymbol{\eta}_2) \\ &\qquad + (1/18)[(5\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\phi}}_2) \times \boldsymbol{\upsilon} - 4\boldsymbol{\alpha} \times (\Delta\dot{\boldsymbol{\phi}}_2 \times \boldsymbol{\upsilon})] \\ &\qquad - (5/18)(\boldsymbol{\alpha} \cdot \Delta\dot{\boldsymbol{\eta}}_2)\boldsymbol{\alpha} + (1/36)\alpha^2\Delta\dot{\boldsymbol{\eta}}_2 \end{aligned} \tag{27}$$

$$\begin{aligned} &\Delta\dot{\boldsymbol{\zeta}}_1 = 0 \qquad \Delta\boldsymbol{\zeta}_1 = 0 \qquad \Delta\dot{\boldsymbol{\zeta}}_2 = \boldsymbol{\upsilon} \qquad \Delta\boldsymbol{\zeta}_2 = \boldsymbol{S}_\upsilon \\ &\Delta\dot{\boldsymbol{\zeta}}_3 = \Delta\boldsymbol{\eta}_2 + (1/6)(\boldsymbol{\alpha} \times \boldsymbol{\upsilon} - 2\boldsymbol{\omega} \times \boldsymbol{S}_\upsilon) \\ &\Delta\dot{\boldsymbol{\zeta}}_4 = \Delta\boldsymbol{\eta}_3 + (1/6)(\Delta\boldsymbol{\phi}_2 \times \boldsymbol{\upsilon} - 2\boldsymbol{\omega} \times \Delta\boldsymbol{\zeta}_3 \\ &\qquad - \Delta\dot{\boldsymbol{\phi}}_2 \times \boldsymbol{S}_\upsilon + \boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\zeta}}_3) \\ &\Delta\dot{\boldsymbol{\zeta}}_5 = \Delta\boldsymbol{\eta}_4 + (1/6)\begin{bmatrix} \boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\zeta}}_4 - 2\boldsymbol{\omega} \times \Delta\boldsymbol{\zeta}_4 - \Delta\dot{\boldsymbol{\phi}}_2 \times \Delta\boldsymbol{\zeta}_3 \\ +\Delta\boldsymbol{\phi}_2 \times \Delta\dot{\boldsymbol{\zeta}}_3 + \Delta\boldsymbol{\phi}_3 \times \boldsymbol{\upsilon} - \Delta\dot{\boldsymbol{\phi}}_3 \times \boldsymbol{S}_\upsilon \end{bmatrix} \\ &\qquad - (1/12)\boldsymbol{\alpha} \times (\Delta\dot{\boldsymbol{\phi}}_2 \times \boldsymbol{S}_\upsilon) + (11/90)(\boldsymbol{\alpha} \times \Delta\dot{\boldsymbol{\phi}}_2) \times \boldsymbol{S}_\upsilon \\ &\qquad + (\alpha^2/60)(\Delta\dot{\boldsymbol{\zeta}}_3 - \Delta\boldsymbol{\eta}_2) - (1/6)\boldsymbol{\alpha}\boldsymbol{\alpha} \cdot (\Delta\dot{\boldsymbol{\zeta}}_3 - \Delta\boldsymbol{\eta}_2) \end{aligned} \tag{28}$$

The Picard expansion equivalent to Eqs. (15) is then formed from Eqs. (20) with Eqs. (19) and (26–28) for $\Delta\dot{\boldsymbol{\chi}}_j$. It is easily verified that under constant angular-rate/specific-force [and the more general conditions noted following Eqs. (16)] that Eqs. (19), (20) and (26–28) reduce to $\dot{\boldsymbol{\phi}} = \Delta\dot{\boldsymbol{\phi}}_1 = \boldsymbol{\omega}$, $\dot{\boldsymbol{\eta}} = \Delta\dot{\boldsymbol{\eta}}_1 = \boldsymbol{a}_{\mathrm{SF}}$, and $\dot{\boldsymbol{\zeta}} = \Delta\dot{\boldsymbol{\zeta}}_2 = \boldsymbol{\upsilon}$, the correct exact solution under these conditions. Thus, under general motion, $\Delta\dot{\boldsymbol{\phi}}_i$, $\Delta\dot{\boldsymbol{\eta}}_i$, $\Delta\dot{\boldsymbol{\zeta}}_{i+1}$ for $i > 1$ measure deviations of $\dot{\boldsymbol{\phi}}$, $\dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$, that is, coning, sculling, and scrolling effects, from the earlier defined conditions.

Potential Continuous-Form High-Speed Algorithms

Strapdown computational algorithms are formulated as the discrete numerical integration equivalent of a continuous differential equation integral. Within the unified framework, computational algorithms are formulated as the digital equivalent of integrated simplified versions of the rotation/translation vector rate equations in Eqs. (15) or the Picard equivalent Eqs. (19) and (20) with Eqs. (26–28). For the remainder of this paper, the simplified versions will be denoted as continuous-form algorithm rate equations. At the conclusion of the paper, we will briefly discuss formulation of algorithms to numerically perform the equivalent integration of the continuous-form rate equations.

The simplest algorithm rate equations (set a) formed from Eqs. (15) neglect all but the leading terms,

$$\dot{\boldsymbol{\phi}}_{\mathrm{Algo/a}} = \boldsymbol{\omega} \qquad \dot{\boldsymbol{\eta}}_{\mathrm{Algo/a}} = \boldsymbol{a}_{\mathrm{SF}} \qquad \dot{\boldsymbol{\zeta}}_{\mathrm{Algo/a}} = \boldsymbol{\upsilon} = \int_{t_{m-1}}^{t} \boldsymbol{a}_{\mathrm{SF}} \,\mathrm{d}t \tag{29}$$

This algorithm set is based on the assumption that in the rotating sensor frame, $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ can be approximated as constants. Under such conditions, Eqs. (15) reduce exactly to the Eqs. (29) form.

Under more dynamic conditions, we might try a truncated form (set b) of Eqs. (15) and (16) with Eqs. (25) such as[12]

$$\begin{aligned} \dot{\boldsymbol{\phi}}_{\mathrm{Algo/b}} &= \boldsymbol{\omega} + \tfrac{1}{2}\boldsymbol{\phi}_{\mathrm{Algo/b}} \times \boldsymbol{\omega} \\ \dot{\boldsymbol{\eta}}_{\mathrm{Algo/b}} &= \boldsymbol{a}_{\mathrm{SF}} + \tfrac{1}{2}(\boldsymbol{\phi}_{\mathrm{Algo/b}} \times \boldsymbol{a}_{\mathrm{SF}} - \dot{\boldsymbol{\phi}}_{\mathrm{Algo/b}} \times \boldsymbol{\eta}_{\mathrm{Algo/b}}) \\ \dot{\boldsymbol{\zeta}}_{\mathrm{Algo/b}} &= \boldsymbol{\eta}_{\mathrm{Algo/b}} + \tfrac{1}{6}(\boldsymbol{\phi}_{\mathrm{Algo/b}} \times \boldsymbol{\eta}_{\mathrm{Algo/b}} - 2\dot{\boldsymbol{\phi}}_{\mathrm{Algo/b}} \times \boldsymbol{\zeta}_{\mathrm{Algo/b}}) \end{aligned} \tag{30}$$

Based on truncated versions of Eqs. (26–28) with Eqs. (19) and (20) [the Picard expansion equivalent to Eqs. (15) and (16)], we could also try the second-order ϕ, η solution and third-order ζ solution (set c):

$$\dot{\phi}_{\text{Algo/c}} = \omega + \tfrac{1}{2}\alpha \times \omega \qquad \dot{\eta}_{\text{Algo/c}} = a_{\text{SF}} + \tfrac{1}{2}(\alpha \times a_{\text{SF}} - \omega \times v)$$
$$\dot{\zeta}_{\text{Algo/c}} = \eta_{\text{Algo/c}} + \tfrac{1}{6}(\alpha \times v - 2\omega \times S_v) \tag{31}$$

The $\dot{\phi}_{\text{Algo/c}}$ approximation in Eqs. (31) was used in Refs. 2, 8, 10, 13–15, and Ref. 7, Sec. 7.1.1.1, where the second term was defined as coning. The second term in $\dot{\eta}_{\text{Algo/c}}$ has been defined as sculling and was the basis for the high-speed portion of two-speed velocity update algorithms in Refs. 6, 8, 15, and Ref. 7, Sec. 7.2.2.2. The second term in $\dot{\zeta}_{\text{Algo/c}}$ is an alternative version of scrolling defined for two-speed high-resolution positioning algorithms (Ref. 6 and Ref. 7, Sec. 7.3.3).

To go with Eqs. (31), we might also try a variation on $\Delta v_{\text{SF}}^{B_{m-1}}$ in Eqs. (17) based on Eq. (4) [and its angular-rate/specific-force approximations described following Eqs. (7)], for the term multiplying η,

$$\begin{aligned}
\eta_{\text{Algo/c}} &= v + \Delta\eta_{\text{Algo/c}} \\
\Delta\eta_{\text{Algo/c}} &\equiv \int_{t_{m-1}}^{t} \frac{1}{2}(\alpha \times a_{\text{SF}} - \omega \times v)\,\mathrm{d}t \\
\Delta v_{\text{SF}_{\text{Algo/d}}}^{B_{m-1}} &= \left[I + f_2(\alpha)(\alpha\times) + f_3(\alpha)(\alpha\times)^2\right]\eta_{\text{Algo/c}} \\
&= v + [f_2(\alpha)I + f_3(\alpha)(\alpha\times)](\alpha \times v) \\
&\quad + \left[I + f_2(\alpha)(\alpha\times) + f_3(\alpha)(\alpha\times)^2\right]\Delta\eta_{\text{Algo/c}} \\
&\approx v + [f_2(\alpha)I + f_3(\alpha)(\alpha\times)](\alpha \times v) + \Delta\eta_{\text{Algo/c}}
\end{aligned} \tag{32}$$

This is the algorithm developed in Ref. 6 and Ref. 7, Secs. 7.2.2.2 and 7.2.2.2-1, which provides the exact solution for $\Delta v_{\text{SF}}^{B_{m-1}}$ under the angular-rate/specific-force conditions described following Eqs. (7). The middle term in the Eqs. (32) $\Delta v_{\text{SF}_{\text{Algo/d}}}^{B_{m-1}}$ expression was defined as exact velocity rotation compensation.

As in Eqs. (32), we might also try a version of $\Delta R_{\text{SF}}^{B_{m-1}}$ in Eqs. (17) based on the original Eq. (5) approximation of constant angular-rate/specific-force for the term multiplying ζ,

$$\begin{aligned}
\zeta_{\text{Algo/c}} &= S_v + \Delta\zeta_{\text{Algo/c}} \\
\Delta\zeta_{\text{Algo/c}} &= \int_{t_{m-1}}^{t} \left[\eta_{\text{Algo/c}} + \frac{1}{6}(\alpha \times v - 2\omega \times S_v)\right]\mathrm{d}t \\
\Delta R_{\text{SF}_{\text{Algo/d}}}^{B_{m-1}} &= \left[I + 2f_3(\alpha)(\alpha\times) + 2f_4(\alpha)(\alpha\times)^2\right]\zeta_{\text{Algo/c}} \\
&= S_v + \left[2f_3(\alpha)(\alpha\times) + 2f_4(\alpha)(\alpha\times)^2\right]S_v \\
&\quad + \left[I + 2f_3(\alpha)(\alpha\times) + 2f_4(\alpha)(\alpha\times)^2\right]\Delta\zeta_{\text{Algo/c}} \\
&\approx S_v + [2f_3(\alpha)I + 2f_4(\alpha)(\alpha\times)](\alpha \times S_v) + \Delta\zeta_{\text{Algo/c}}
\end{aligned} \tag{33}$$

A variation on Eqs. (33) is based on the following development:

$$S\alpha \equiv \int_{t_{m-1}}^{t} \alpha\,\mathrm{d}t$$

$$\Delta\dot{\zeta}_{\text{Algo/c}} = \Delta\eta_{\text{Algo/c}} + \frac{1}{6}(\alpha \times v - 2\omega \times S_v) = \Delta\eta_{\text{Algo/c}}$$
$$+ \frac{1}{6}\left[\alpha \times v - S_\alpha \times a_{\text{SF}} - \omega \times S_v + \frac{\mathrm{d}}{\mathrm{d}t}(S_\alpha \times v - \alpha \times S_v)\right]$$

$$\Delta\zeta_{\text{Algo/d}} \equiv \int_{t_{m-1}}^{t} \left[\Delta\eta_{\text{Algo/c}} + \frac{1}{6}(\alpha \times v - S_\alpha \times a_{\text{SF}} - \omega \times S_v)\right]\mathrm{d}t$$

$$\zeta_{\text{Algo/c}} = S_v + \Delta\zeta_{\text{Algo/c}} = S_v + \frac{1}{6}(S_\alpha \times v - \alpha \times S_v) + \Delta\zeta_{\text{Algo/d}} \tag{34}$$

Substitution in Eqs. (33), approximating $\frac{1}{6} \approx f_3(\alpha)$, and recognizing $\alpha f_4(\alpha)$ to be small compared to $f_3(\alpha)$ so that $f_3(\alpha)I \approx f_3(\alpha)I + f_4(\alpha)(\alpha\times)$ then yields

$$\begin{aligned}
\Delta R_{\text{SF}_{\text{Algo/d}}}^{B_{m-1}} &= \left[I + 2f_3(\alpha)(\alpha\times) + 2f_4(\alpha)(\alpha\times)^2\right]\Big(S_v \\
&\quad + \tfrac{1}{6}(S_\alpha \times v - \alpha \times S_v) + \Delta\zeta_{\text{Algo/d}}\Big) \\
&\approx S_v + [2f_3(\alpha)I + 2f_4(\alpha)(\alpha\times)](\alpha \times S_v) \\
&\quad + f_3(\alpha)(S_\alpha \times v - \alpha \times S_v) + \Delta\zeta_{\text{Algo/d}} \\
&\approx S_v + 2[f_3(\alpha)I + f_4(\alpha)(\alpha\times)](\alpha \times S_v) + [f_3(\alpha)I \\
&\quad + f_4(\alpha)(\alpha\times)](S_\alpha \times v - \alpha \times S_v) + \Delta\zeta_{\text{Algo/d}} \\
&= S_v + [f_3(\alpha)I + f_4(\alpha)(\alpha\times)](S_\alpha \times v \\
&\quad + \alpha \times S_v) + \Delta\zeta_{\text{Algo/d}}
\end{aligned} \tag{35}$$

Note that under constant angular-rate/specific-force, the $\frac{1}{6}(S_\alpha \times v - \alpha \times S_v)$ term in Eq. (35) is identically zero; hence, approximations for this term in Eq. (35) have no impact on $\Delta R_{\text{SF}_{\text{Algo/d}}}^{B_{m-1}}$ accuracy under such conditions. The Eq. (35) result is the algorithm developed in Ref. 6 and Ref. 7, Secs. 7.3.3 and 7.3.3.1, that provides the exact solution for $\Delta R_{\text{SF}}^{B_{m-1}}$ under constant angular-rate/specific-force. The middle term in Eq. (35) was defined as exact position rotation compensation and $\Delta\zeta_{\text{Algo/d}}$ as computed in Eqs. (34) was defined as the scrolling term for high-resolution position updating in a dynamic environment.

Algorithm Accuracy Assessment

The accuracy of the $\Delta v_{\text{SF}}^{B_{m-1}}$ and $\Delta R_{\text{SF}}^{B_{m-1}}$ algorithms in Eqs. (32), (33), and (35) is limited by the approximations of using α for ϕ and not including $\Delta\eta_{\text{Algo/c}}$ and $\Delta\zeta_{\text{Algo/c}}$ in the middle rotation compensation terms. No further discussion of the effect on accuracy is warranted because the associated error is easily eliminated by using the Eqs. (17) form directly.

For the $\dot{\phi}$, $\dot{\eta}$, and $\dot{\zeta}$ continuous-form algorithms in Eqs. (29–31), the error can be calculated as the algorithm output minus the equivalent true output represented by Eqs. (19), (20), and (26–28). Thus, the error in Eqs. (29) algorithm set a is approximately the negative of $\Delta\dot{\phi}_2$, $\Delta\dot{\eta}_2$, and $\Delta\dot{\zeta}_3$ in Eqs. (26–28),

$$\delta\dot{\phi}_{\text{Algo/a}} \approx -\Delta\dot{\phi}_2 \qquad \delta\dot{\eta}_{\text{Algo/a}} \approx -\Delta\dot{\eta}_2 \qquad \delta\dot{\zeta}_{\text{Algo/a}} \approx -\Delta\dot{\zeta}_3 \tag{36}$$

As expected, the algorithm rate errors are first order for $\delta\dot{\phi}$ and $\delta\dot{\eta}$ and second order for $\delta\dot{\zeta}$, that is, second-order errors in $\delta\phi$ and $\delta\eta$ and third-order errors in $\delta\zeta$.

To assess the accuracy of Eqs. (30) algorithm rate set b, we develop an order higher Picard iterative solution for Eqs. (30) [similar to the approach used to generate Eqs. (26–28)] and compare the solution with the equivalent of Eqs. (19), (20), and (26–28). The result is

$$\begin{aligned}
\delta\dot{\phi}_{\text{Algo/b}} &\approx -\tfrac{1}{6}\alpha \times \Delta\dot{\phi}_2 \\
\delta\dot{\eta}_{\text{Algo/b}} &\approx -\tfrac{1}{6}(\alpha \times \Delta\dot{\eta}_2 + 2\Delta\dot{\phi}_2 \times v) \\
\delta\dot{\zeta}_{\text{Algo/b}} &\approx \delta\eta_{\text{Algo/b}} - \tfrac{1}{6}[\Delta\dot{\phi}_2 \times S_v + \alpha \times (\Delta\dot{\zeta}_3 - \Delta\eta_2)]
\end{aligned} \tag{37}$$

As expected, the algorithm rate errors are second order in $\delta\dot{\phi}$ and $\delta\dot{\eta}$ and third order in $\delta\dot{\zeta}$, that is, third-order errors in $\delta\phi$ and $\delta\eta$ and fourth-order errors in $\delta\zeta$.

Because Eqs. (31) algorithm set c was formed as $\dot{\phi}_2$, $\dot{\eta}_2$, $\dot{\zeta}_3$, we expect the error rate in set c to be approximately the negative of $\Delta\dot{\phi}_3$, $\Delta\dot{\eta}_3$, and $\Delta\dot{\zeta}_4$ in Eqs. (26–28),

$$\delta\dot{\phi}_{\text{Algo/c}} \approx -\Delta\dot{\phi}_3 \qquad \delta\dot{\eta}_{\text{Algo/c}} \approx -\Delta\dot{\eta}_3 \qquad \delta\dot{\zeta}_{\text{Algo/c}} \approx -\Delta\dot{\zeta}_4 \tag{38}$$

On review, the set c error rates in Eqs. (38) appear to be second order for $\delta\dot{\phi}$ and $\delta\dot{\eta}$ and third order for $\delta\dot{\zeta}$ as expected. However, on further analysis, it can be demonstrated that based on a time-ordered expansion (in contrast with the $\boldsymbol{\alpha}$-, $\boldsymbol{\upsilon}$-, and $\boldsymbol{\zeta}$-ordered Picard expansion discussed thus far), Eqs. (38) actually have one order smaller errors in $\delta\dot{\phi}$, $\delta\dot{\eta}$ than would be expected. To demonstrate, we represent the $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\text{SF}}$ inputs to Eqs. (38) by the time-ordered Taylor series expansion,

$$\begin{aligned}
\boldsymbol{\omega} &= \boldsymbol{\omega}_0 + \dot{\boldsymbol{\omega}}_0(t - t_{m-1}) + \ddot{\boldsymbol{\omega}}_0\big[(t - t_{m-1})^2/2!\big] \\
&\quad + \dddot{\boldsymbol{\omega}}_0\big[(t - t_{m-1})^3/3!\big] + \cdots \\
\boldsymbol{a}_{\text{SF}} &= \boldsymbol{a}_0 + \dot{\boldsymbol{a}}_0(t - t_{m-1}) + \ddot{\boldsymbol{a}}_0\big[(t - t_{m-1})^2/2!\big] \\
&\quad + \dddot{\boldsymbol{a}}_0\big[(t - t_{m-1})^3/3!\big] + \cdots
\end{aligned} \tag{39}$$

for which

$$\begin{aligned}
\boldsymbol{\alpha} &= \boldsymbol{\omega}_0(t - t_{m-1}) + \dot{\boldsymbol{\omega}}_0\big[(t - t_{m-1})^2/2!\big] \\
&\quad + \ddot{\boldsymbol{\omega}}_0\big[(t - t_{m-1})^3/3!\big] + \cdots \\
\boldsymbol{\upsilon} &= \boldsymbol{a}_0(t - t_{m-1}) + \dot{\boldsymbol{a}}_0\big[(t - t_{m-1})^2/2!\big] \\
&\quad + \ddot{\boldsymbol{a}}_0\big[(t - t_{m-1})^3/3!\big] + \cdots \\
\boldsymbol{S}_\upsilon &= \boldsymbol{a}_0\big[(t - t_{m-1})^2/2!\big] + \dot{\boldsymbol{a}}_0\big[(t - t_{m-1})^3/3!\big] \\
&\quad + \ddot{\boldsymbol{a}}_0\big[(t - t_{m-1})^4/4!\big] + \cdots
\end{aligned} \tag{40}$$

On substitution of Eqs. (39) and (40) in Eqs. (38), we find for particular terms

$$\begin{aligned}
\Delta\dot{\boldsymbol{\phi}}_2 &= (1/4)\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0(t - t_{m-1})^2 + \cdots \\
\Delta\boldsymbol{\phi}_2 &= (1/12)\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0(t - t_{m-1})^3 + \cdots \\
\Delta\dot{\boldsymbol{\eta}}_2 &= (1/4)(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - \dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0)(t - t_{m-1})^2 + \cdots \\
\Delta\boldsymbol{\eta}_2 &= (1/12)(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - \dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0)(t - t_{m-1})^3 + \cdots
\end{aligned} \tag{41}$$

Because $\Delta\dot{\boldsymbol{\phi}}_2$ and $\Delta\boldsymbol{\phi}_2$ in Eqs. (41) are second and third order in $(t - t_{m-1})$ powers, respectively, we would expect $\Delta\dot{\boldsymbol{\phi}}_3$ in Eqs. (38) to be third order in $(t - t_{m-1})$. Similarly for $\Delta\dot{\boldsymbol{\eta}}_2$ and $\Delta\boldsymbol{\eta}_2$ in Eqs. (41) and $\Delta\dot{\boldsymbol{\eta}}_3$ in Eqs. (38). However, from Eqs. (41), we also see that

$$\begin{aligned}
\Delta\dot{\boldsymbol{\phi}}_2 &= [3/(t - t_{m-1})]\Delta\boldsymbol{\phi}_2 + \cdots \\
\Delta\dot{\boldsymbol{\eta}}_2 &= [3/(t - t_{m-1})]\Delta\boldsymbol{\eta}_2 + \cdots
\end{aligned} \tag{42}$$

On substituting Eqs. (42) in the Eqs. (38) $\Delta\dot{\boldsymbol{\phi}}_3$ and $\Delta\dot{\boldsymbol{\eta}}_3$ equations, we discover that the third-order $(t - t_{m-1})$ terms cancel so that $\Delta\dot{\boldsymbol{\phi}}_3$ and $\Delta\dot{\boldsymbol{\eta}}_3$ are actually accurate to fourth order. Thus, $\delta\dot{\boldsymbol{\phi}}_{\text{Algo/c}}$ and $\delta\dot{\boldsymbol{\eta}}_{\text{Algo/c}}$ in Eqs. (38) are one order more accurate than originally expected, that is, fourth order in powers of $(t - t_{m-1})$. Hence, $\dot{\boldsymbol{\phi}}_{\text{Algo/c}}$ and $\dot{\boldsymbol{\eta}}_{\text{Algo/c}}$ in Eqs. (31), which are first order in $\boldsymbol{\alpha}$, $\boldsymbol{\upsilon}$ products, are more accurate in maneuvering environments than might be expected from first-order algorithms.

Continuous-Form Algorithm Approximation Errors

In this section, we determine analytical expressions for evaluating the errors remaining in algorithm sets a, b, and c as a function of generalized angular-rate/specific-force input maneuver and vibration profile characteristics.

Algorithm Errors Under Maneuvers

Maneuver profiles can be characterized in general by the Taylor series expansion forms of Eqs. (39). Under Eqs. (39) maneuver inputs, Eqs. (36) and (37) were used to assess the $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ errors in algorithm sets a and b, and Equations (38) were used to assess the $\boldsymbol{\zeta}$ algorithm set c error. As discussed in the preceding section, the $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$ errors for algorithm set c have higher than expected accuracy based on generalized Eqs. (39) maneuver profile characteristics. As such, rather than Eqs. (38), the following more accurate equations were used for evaluating the $\boldsymbol{\phi}$, $\boldsymbol{\eta}$ algorithm set c errors under maneuvering conditions:

$$\begin{aligned}
\delta\dot{\boldsymbol{\phi}}_{\text{Algo/c}} &= \dot{\boldsymbol{\phi}}_{\text{Algo/c}} - \sum_1^4 \Delta\dot{\boldsymbol{\phi}}_i = -\Delta\dot{\boldsymbol{\phi}}_3 - \Delta\dot{\boldsymbol{\phi}}_4 \\
\delta\dot{\boldsymbol{\eta}}_{\text{Algo/c}} &= \dot{\boldsymbol{\eta}}_{\text{Algo/c}} - \sum_1^4 \Delta\dot{\boldsymbol{\eta}}_i = -\Delta\dot{\boldsymbol{\eta}}_3 - \Delta\dot{\boldsymbol{\eta}}_4
\end{aligned} \tag{43}$$

To evaluate the $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ errors in terms of maneuver profile parameters, substitute Eqs. (39) and (40) in Eqs. (26–28), the result into Eqs. (36–38) and (43) as discussed earlier and analytically integrate to obtain $\delta\boldsymbol{\phi}$, $\delta\boldsymbol{\eta}$, and $\delta\boldsymbol{\zeta}$. After much routine algebra, the final result is

$$\begin{aligned}
\delta\boldsymbol{\phi}_{\text{Algo/a}_m} &= -(1/12)\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0 T_m^3 + \cdots = -\boldsymbol{\phi}_{\text{cone}_m} \\
\delta\boldsymbol{\phi}_{\text{Algo/b}_m} &= -(1/96)\boldsymbol{\omega}_0 \times (\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) T_m^4 + \cdots \\
\delta\boldsymbol{\phi}_{\text{Algo/c}_m} &= -(1/720)\big[3(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) \times \dot{\boldsymbol{\omega}}_0 - (\boldsymbol{\omega}_0 \times \ddot{\boldsymbol{\omega}}_0) \\
&\quad \times \boldsymbol{\omega}_0 + \omega_0^2(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0)\big] T_m^5 + \cdots
\end{aligned} \tag{44}$$

$$\begin{aligned}
\delta\boldsymbol{\eta}_{\text{Algo/a}_m} &= -(1/12)(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - \dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) T_m^3 + \cdots = -\boldsymbol{\eta}_{\text{scull}_m} \\
\delta\boldsymbol{\eta}_{\text{Algo/b}_m} &= -(1/96)[\boldsymbol{\omega}_0 \times (\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - \dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) \\
&\quad + 2(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) \times \boldsymbol{a}_0] T_m^4 + \cdots \\
\delta\boldsymbol{\eta}_{\text{Algo/c}_m} &= -\frac{1}{720}\begin{bmatrix} \big[\omega_0^2 I - 3(\dot{\boldsymbol{\omega}}_0 \times)\big](\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - \dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) \\ -(\boldsymbol{\omega}_0 \times \ddot{\boldsymbol{\omega}}_0) \times \boldsymbol{a}_0 - (\boldsymbol{\omega}_0 \times \ddot{\boldsymbol{a}}_0 - \ddot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) \times \boldsymbol{\omega}_0 \\ +[2(\boldsymbol{a}_0 \cdot \boldsymbol{\omega}_0) I - 3(\dot{\boldsymbol{a}}_0 \times)](\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) \end{bmatrix} T_m^5 + \cdots
\end{aligned} \tag{45}$$

$$\begin{aligned}
\delta\boldsymbol{\zeta}_{\text{Algo/a}_m} &= -(1/72)(2\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - 3\dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) T_m^4 + \cdots = -\boldsymbol{\zeta}_{\text{scroll}_m} \\
\delta\boldsymbol{\zeta}_{\text{Algo/b}_m} &= -(1/4320)\{\boldsymbol{\omega}_0 \times [13(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0) - 21(\dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0)] \\
&\quad + 36(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) \times \boldsymbol{a}_0\} T_m^5 + \cdots \\
\delta\boldsymbol{\zeta}_{\text{Algo/c}_m} &= -(1/4320)[\boldsymbol{\omega}_0 \times (8\boldsymbol{\omega}_0 \times \dot{\boldsymbol{a}}_0 - 12\dot{\boldsymbol{\omega}}_0 \times \boldsymbol{a}_0) \\
&\quad - 6(\boldsymbol{\omega}_0 \times \dot{\boldsymbol{\omega}}_0) \times \boldsymbol{a}_0] T_m^5 + \cdots
\end{aligned} \tag{46}$$

From Eqs. (44–46), we see, as expected, that algorithm a (which approximates the coning, sculling, and scrolling effects as negligible) is in error by minus the coning, sculling, and scrolling terms. The algorithm c error result for $\boldsymbol{\phi}$ in Eqs. (44) matches that derived by a similar method in Ref. 14, providing confidence in results obtained. For the $\boldsymbol{\phi}$, $\boldsymbol{\eta}$ algorithm b vs algorithm c errors, it is obvious that the algorithm c solution is superior because, as already discussed, its

errors are fifth order in T_m vs algorithm b, whose errors are fourth order in T_m. Note also that the ζ error for algorithm c is smaller than for algorithm b based on the magnitude of coefficients for like terms. We can conclude that for maneuver-type profiles characterized by Eqs. (39), algorithm c is more accurate than algorithm b.

Algorithm Errors Under Vibration

For analysis of algorithm errors under vibration, the following sinusoidal model was used to characterize angular-rate/specific-force:

$$\boldsymbol{\omega} = \theta_0\Omega\cos\Omega t\boldsymbol{u}_x + \theta_0\Omega\sin\Omega t\boldsymbol{u}_y \qquad \boldsymbol{a}_{\mathrm{SF}} = a_{\mathrm{SF}_0}\sin\Omega t\boldsymbol{u}_y \quad (47)$$

for which

$$\boldsymbol{\alpha} = \theta_0(\sin\Omega t - \sin\Omega t_{m-1})\boldsymbol{u}_x + \theta_0(\cos\Omega t_{m-1} - \cos\Omega t)\boldsymbol{u}_y$$

$$\boldsymbol{\upsilon} = a_{\mathrm{SF}_0}(1/\Omega)(\cos\Omega t_{m-1} - \cos\Omega t)\boldsymbol{u}_y$$

$$\boldsymbol{S}_\upsilon = a_{\mathrm{SF}_0}(1/\Omega^2)[\Omega(t - t_{m-1})\cos\Omega t_{m-1} - (\sin\Omega t - \sin\Omega t_{m-1})]\boldsymbol{u}_y \quad (48)$$

To evaluate the ϕ, η, and ζ errors in terms of the Eqs. (47) and (48) vibration profile parameters, substitute Eqs. (47) and (48) in Eqs. (26–28), the result into Eqs. (36–38), and analytically integrate to obtain $\delta\phi$, $\delta\eta$, and $\delta\zeta$. After much routine algebra, the final result is

$$\delta\boldsymbol{\phi}_{\mathrm{Algo/a}_m} = -(1/2)\theta_0^2(\beta - \sin\beta)\boldsymbol{u}_z = -\boldsymbol{\phi}_{\mathrm{cone}_m}$$

$$\delta\boldsymbol{\phi}_{\mathrm{Algo/b}_m} = (1/24)\theta_0^3[g_1\Delta S(\Omega)_m + 3(\beta - \sin\beta)\sin(\beta/2)\Delta C(\Omega)_m]\boldsymbol{u}_x - (1/24)\theta_0^3[g_1\Delta C(\Omega)_m - 3(\beta - \sin\beta)\sin(\beta/2)\Delta S(\Omega)_m]\boldsymbol{u}_y$$

$$\delta\boldsymbol{\phi}_{\mathrm{Algo/c}_m} = (1/24)\theta_0^3 4g_1\Delta S(\Omega)_m\boldsymbol{u}_x - (1/24)\theta_0^3 4g_1\Delta C(\Omega)_m\boldsymbol{u}_y \quad (49)$$

$$\delta\boldsymbol{\eta}_{\mathrm{Algo/a}_m} = -(1/2)\theta_0\, a_{\mathrm{SF}_0}(1/\Omega)(\beta - \sin\beta)\boldsymbol{u}_z = -\boldsymbol{\eta}_{\mathrm{scull}_m}$$

$$\delta\boldsymbol{\eta}_{\mathrm{Algo/b}_m} = -(1/24)\theta_0^2\, a_{\mathrm{SF}_0}(1/\Omega)[g_1\Delta S(\Omega)_m + 3(\beta - \sin\beta)\sin(\beta/2)\Delta C(\Omega)_m]\boldsymbol{u}_x - (1/24)\theta_0^2\, a_{\mathrm{SF}_0}(1/\Omega)[g_1\Delta C(\Omega)_m - 3(\beta - \sin\beta)\sin(\beta/2)\Delta S(\Omega)_m]\boldsymbol{u}_y$$

$$\delta\boldsymbol{\eta}_{\mathrm{Algo/c}_m} = (1/24)\theta_0^2\, a_{\mathrm{SF}_0}(1/\Omega)8g_1\Delta S(\Omega)_m\boldsymbol{u}_x - (1/24)\theta_0^2\, a_{\mathrm{SF}_0}(1/\Omega)4g_1\Delta C(\Omega)_m\boldsymbol{u}_y \quad (50)$$

$$\delta\boldsymbol{\zeta}_{\mathrm{Algo/a}_m} = -(1/12)\theta_0\, a_{\mathrm{SF}_0}(1/\Omega^2)[g_2 + g_3\Delta S(2\Omega)_m + g_4\Delta C(2\Omega)_m]\boldsymbol{u}_z = -\boldsymbol{\zeta}_{\mathrm{scroll}_m} \quad (51)$$

for which the following definitions apply:

$$\Delta S(\rho)_m \equiv \frac{\sin\rho t_m - \sin\rho t_{m-1}}{2\sin\frac{1}{2}\rho T_m} \qquad \Delta C(\rho)_m \equiv \frac{\cos\rho t_m - \cos\rho t_{m-1}}{2\sin\frac{1}{2}\rho T_m}$$

$$\beta \equiv \Omega T_m$$

$$g_1 \equiv 4\sin\tfrac{1}{2}\beta - (3\beta - \sin\beta)\cos\tfrac{1}{2}\beta$$

$$g_2 \equiv 3\beta^2 - 2(1 - \cos\beta + \beta\sin\beta)$$

$$g_3 \equiv \beta\sin\beta - 2(1 - \cos\beta)$$

$$g_4 \equiv -3\sin\beta + \beta(2 + \cos\beta) \quad (52)$$

and where ρ is a generalized frequency parameter. As an exercise, the curious reader may wish to derive equations for $\delta\boldsymbol{\zeta}_{\mathrm{Algo/b}_m}$ and $\delta\boldsymbol{\zeta}_{\mathrm{Algo/c}_m}$ to go with $\delta\boldsymbol{\zeta}_{\mathrm{Algo/a}_m}$ in Eqs. (52). Because $\boldsymbol{\zeta}_{\mathrm{Algo/a}}$ is acceptable for almost all applications [based on numerical evaluation of Eq. (51)], it is believed that either $\boldsymbol{\zeta}_{\mathrm{Algo/b}}$ or $\boldsymbol{\zeta}_{\mathrm{Algo/c}}$ has sufficient accuracy to satisfy accuracy requirements for the few remaining high-precision positioning applications and that accuracy differences between the two should be a nonfactor.

From Eqs. (49–51), we see, as expected, that algorithm a (which approximates the coning, sculling, and scrolling effects as negligible) is in error by minus the coning, sculling, and scrolling terms.

Algorithms b and c for ϕ and η and algorithm a for ζ contain ΔS_m and ΔC_m terms that need interpretation. First, we note that the effect of $\delta\boldsymbol{\phi}_{\mathrm{Algo}_m}$, $\delta\boldsymbol{\eta}_{\mathrm{Algo}_m}$, and $\delta\boldsymbol{\zeta}_{\mathrm{Algo}_m}$ error is to accumulate into attitude, velocity, and position error at the m cycle rate. For slow maneuvers, the resulting errors can be approximated as the sum of $\delta\boldsymbol{\phi}_{\mathrm{Algo}_m}$, $\delta\boldsymbol{\eta}_{\mathrm{Algo}_m}$, and $\delta\boldsymbol{\zeta}_{\mathrm{Algo}_m}$ over a succession of m cycles, for example, from $m = 1$ to M beginning at time t_0. In Eqs. (49–51), ΔS_m and ΔC_m are the only parameters that change with m; hence, summing $\delta\boldsymbol{\phi}_{\mathrm{Algo}_m}$, $\delta\boldsymbol{\eta}_{\mathrm{Algo}_m}$, and $\delta\boldsymbol{\zeta}_{\mathrm{Algo}_m}$ is equivalent to summing ΔS_m and ΔC_m with appropriate multiplication coefficients used to evaluate cumulative attitude, velocity, and position error. From the definition of ΔS_m and ΔC_m in Eqs. (52), the sums are

$$\sum_1^M \Delta S(\rho)_m = \frac{\sin\rho t_M - \sin\rho t_0}{2\sin\frac{1}{2}\rho T_M}$$

$$\sum_1^M \Delta C(\rho)_m = \frac{\cos\rho t_M - \cos\rho t_0}{2\sin\frac{1}{2}\rho T_M} \quad (53)$$

Now consider an aliased (folded) frequency parameter ρ' defined implicitly by $\rho T_m = 2K\pi + \rho' T_m$, where K is the nearest integer number of ρ-frequency cycles in an m cycle. Substitution finds for Eqs. (53) terms

$$\rho t_M = \rho t_0 + \rho M T_m = \rho t_0 + M(2K\pi + \rho' T_m) = 2MK\pi + \rho t_0 + \rho'(t_m - t_0)$$

$$2\sin\tfrac{1}{2}\rho T_M = 2\cos K\pi\sin\tfrac{1}{2}\rho' T m = \rho' T_m\cos K\pi\frac{\sin\frac{1}{2}\rho' T_m}{\frac{1}{2}\rho' T_m} = \rho' T_m\cos K\pi f_1\left(\tfrac{1}{2}\rho' T_m\right)$$

where $f_1(\,)$ is the Taylor series defined in Eqs. (8). Then Eqs. (53) become

$$\sum_1^M \Delta S(\rho)_m = \int_{t_0}^{t_M} G_1(t)\,\mathrm{d}t \qquad \sum_1^M \Delta C(\rho)_m = \int_{t_0}^{t_M} G_2(t)\,\mathrm{d}t$$

$$G_1(t) = J(\rho')\cos[\rho t_0 + M\rho'(t - t_0)]$$

$$G_2(t) = -J(\rho')\sin[\rho t_0 + M\rho'(t - t_0)]$$

$$J(\rho') = \frac{1}{T_m\cos K\pi f_1\left(\frac{1}{2}\rho' T_m\right)} \quad (54)$$

Equations (54) show that summing ΔS_m and ΔC_m generates sinusoidal waveforms at the folding frequency ρ' that represent the integral of sinusoids of frequency ρ' and amplitude $J(\rho')$. Note that $J(\rho')$ has no singularities for any ρ' value over its defining range from $-\pi$ to $+\pi$. At $\rho' = 0$, G_1 and G_2 become constant at $J(0)\cos\rho t_0$ and $-J(0)\sin\Omega t_0$, and the ΔS_m, ΔC_m sums increase linearly with t_M, the limit condition for the integral of sinusoids at frequency $\rho' \to 0$ having an initial phase angle of ρt_0.

The preceding exercise can now be used to interpret the ΔS_m and ΔC_m terms in Eqs. (49–51) as they impact attitude, velocity, and position error. For example, consider $\delta\boldsymbol{\phi}_{\mathrm{Algo/c}_m}$ in Eqs. (49). The process of summing $\delta\boldsymbol{\phi}_{\mathrm{Algo/c}_m}$ into attitude error is the Eqs. (54) effect multiplied by the Eqs. (49) coefficient $(1/24)\theta_0^3 4g_1$. Hence, the resulting attitude error is the integral of a sinusoid at folding frequency Ω' with amplitude $(1/24)\theta_0^3 4g_1 J(\Omega')$, or with Eqs. (54)

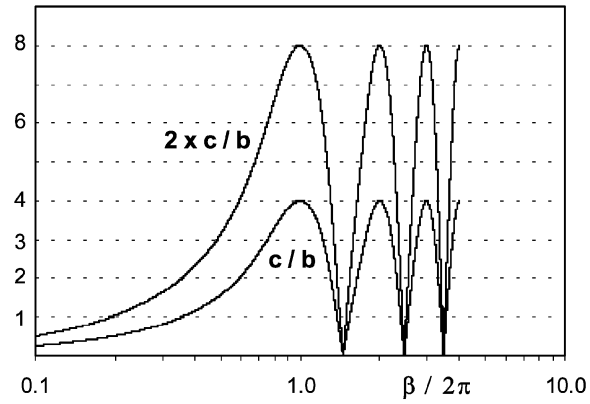


Fig. 1 Algorithm c vs algorithm b error in ϕ and η under vibration.

for $J(\Omega')$, the equivalent to the integral of a sinusoidal angular-rate vibration of amplitude $\theta_0^3 g_1/6T_m f_1(\frac{1}{2}\Omega' T_m)$ at frequency Ω'. For $\delta\boldsymbol{\eta}_{\text{Algo/c}_m}$ in Eqs. (50), the resulting y component velocity error is equivalent to the integral of a sinusoidal acceleration vibration of amplitude $\theta_0^2 a_{\text{SF}_0} g_1/6\Omega T_m f_1(\frac{1}{2}\Omega' T_m)$ and frequency Ω'.

The preceding discussion showed that the effect of vibration on $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$ algorithms b and c is to generate a sinusoidal error in attitude and velocity at the folding frequency Ω'. A comparison of the $\boldsymbol{\phi}$ and $\boldsymbol{\eta}$ algorithm b and c error equations in Eqs. (49) and (50) also reveals some interesting characteristics as a function of the vibration frequency parameter $\beta = \Omega T_m$. Both the algorithms b and c errors contain the g_1 term (a function of β); however, the magnitude is four times higher for algorithms c (actually eight times higher for the x component of $\boldsymbol{\eta}$ error). On the other hand, algorithm b errors contain an additional β function $3(\beta - \sin\beta)\sin(\beta/2)$ not contained in the algorithm c error. It can be shown that the combined effect of g_1 and $3(\beta - \sin\beta)\sin(\beta/2)$ on algorithm b per axis error magnitude is given by $\sqrt{\{g_1^2 + [3(\beta - \sin\beta)\sin(\beta/2)]^2\}}$. Thus, the ratio of algorithm c to algorithm b error can be analyzed as the ratio of $4|g_1|$ (or $8\ |g_1|$) for the algorithm c error divided by the previous algorithm b function: algorithms c/b error ratio $= 4|g_1|/\sqrt{\{g_1^2 + [3(\beta - \sin\beta)\sin(\beta/2)]^2\}}$. Figure 1 shows this ratio as a function of $\beta/2\pi$ (which equals 1 when the vibration frequency in hertz equals the m cycle frequency). Also shown in Fig. 1 is two times the same ratio corresponding to the 8 g_1 error effect for the x component of the $\boldsymbol{\eta}$ algorithm c error noted earlier.

Figure 1 shows that the error in algorithm c is smaller than for algorithm b for vibration frequencies lower than 0.35 times the m cycle update frequency (and lower than 0.2 times the m cycle update frequency for the $2\times$ algorithms c/b error ratio). Otherwise, algorithm b errors are smaller, except for small narrow regions at 1.5, 2.5, 3.5, etc., times the m cycle frequency for which algorithm c errors are again smaller. Because $\boldsymbol{\phi}$, $\boldsymbol{\eta}$ algorithms b and c errors are cyclic with typically small amplitudes, the preceding discussion may be of only academic interest.

Digital Algorithm Development

Digital algorithms would be used in a strapdown system computer to implement integration of the rotation and velocity/position translation vector rate equations. The process is complicated by the form of input data typically provided by the strapdown angular-rate sensors and accelerometers, increments of integrated angular rate and specific force acceleration over the time period for input data sampling. Selection of the input data sampling period is part of the digital algorithm design process and is based on accuracy considerations in expected dynamic environments. In general, two methods can be considered.

1) Divide the m cycle into a number of increments corresponding to the order of digital algorithm to be used for the digital integration.[12] Input sensor integrated increments for these samples and generate an integral solution for the rotation/translation vectors over the m cycle using the samples.

2) Perform the rotation/translation vector digital integrations at a higher speed (l cycle rate) compared to the m cycle rate. Base the l cycle integration updates on sensor samples taken at the l cycle rate or at a higher rate within the l cycle. The number of sensor samples used for the integration update would correspond to the order of the digital algorithm being implemented (Refs. 1, 2, 8, 10, 13, 15 and Ref. 7, Secs. 7.1.1.1.1, 7.2.2.2.2, and 7.3.3.2). The sensor samples used for l cycle updates could be from current and past l cycles, from higher rate samples within an l cycle, or a combination thereof.

For approach 1, the sensor sampling/processing rate is determined by the digital integration algorithm order and the m rate. The m rate would be set high enough to generate sufficient sensor inputs for an accurate navigation solution under high-frequency dynamic input conditions. For approach 2, the m rate is designed to maintain a small value of ϕ under maximum angular rate conditions, thereby protecting approximations in the continuous form algorithms (e.g., 100 Hz to maintain ϕ less than 0.04 rad under a maximum 4-rad/s angular rate). High-frequency dynamic inputs (i.e., vibration that is generally of small ϕ amplitude) is accurately processed by setting a high enough l cycle rate (e.g., 3 kHz to measure random vibrations from 20 to 500 Hz). Thus, approach 2 benefits from the advantage afforded by a two-speed algorithm structure when computer throughput limitations are an issue; only the l rate must increase under high-frequency vibration to maintain accurate navigation; the m rate and its updating operations can be maintained at a lower frequency without impacting accuracy.

Simulation Testing for Equation Validation

The analytical design process used in preparing this paper included several rederivations of all analytical results by the author and independently by K. M. Roscoe of Applied Strapdown Analytics, until both derivations agreed. Because of the complexity of the analytical results, digital simulation testing was also conducted for added confidence. The simulation tests were constructed to provide a numerical accuracy assessment of the key derived equations compared with reference numerical solutions of higher or equal accuracy. Tests were designed to verify exact solution equations (15–17), Picard solution equations (20) with Eqs. (26–28), maneuver-induced error equations (44–46), and vibration-induced error equations (49–51). The validation process and results obtained are described in the Appendix.

Conclusions

The velocity and position updating function in strapdown inertial navigation systems can be structured using a new two-speed formulation that directly parallels the Jordan/Bortz/Laning approach for attitude updating. The new velocity/position update routines generate an exact solution when provided with exact position/velocity translation vector inputs (directly analogous to the rotation vector input for Jordan/Bortz/Laning attitude updating). Differential equations have been derived whose integration over a velocity/position update cycle provide the exact translation vectors (analogous to the Laning/Bortz rotation vector rate equation). Digital integration algorithms can be designed for the translation vectors using the exact differential equations (or their Picard expansion equivalent) as a design base. This is the approach that has been successfully used in the past for rotation vector algorithm design based on the Jordan/Bortz/Laning rotation vector rate equation. Under constant strapdown angular-rate/specific-force, the translation vectors reduce identically to the first and second integral of strapdown specific force acceleration (analogous to the rotation vector, which reduces to the integral of strapdown angular rate under such conditions). Because the attitude/velocity/position updating routines are exact under constant input conditions, their software equivalents can be easily and precisely validated by simulation under these conditions and comparison with known closed-form solutions. Because the updating equations are also exact under general motion, once validated, their algorithmic form can be used for all applications. Modern day computer capabilities allow the use of such exact strapdown updating algorithms without penalty.

Appendix: Equation Validation

This Appendix describes the digital simulation tests used to validate exact solution equations (15–17), Picard solution equations (20) with Eqs. (26–28), maneuver-induced error equations (44–46), and vibration-induced error equations (49–51). For all tests, integral solutions for the navigation parameters were generated using a trapezoidal integration algorithm with a 0.1-μs cycle time. Depending on the test, the m cycle time period was set to 0.1 or 0.01 s, that is, a 10- or 100-Hz navigation parameter update rate.

Exact Solution Validation

For this test, the exact solution for $C_{B_m}^{B_{m-1}}$, $\Delta \boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}}$, and $\Delta \boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}}$ obtained using Eqs. (15–17) by integrating $\dot{\boldsymbol{\phi}}$, $\dot{\boldsymbol{\eta}}$, and $\dot{\boldsymbol{\zeta}}$ was compared with an exact reference solution generated by direct integration of $\dot{C}_{B(t)}^{B_{m-1}}$, $\Delta \dot{\boldsymbol{v}}_{\mathrm{SF}}^{B_{m-1}}(t)$, and $\Delta \dot{\boldsymbol{R}}_{\mathrm{SF}}^{B_{m-1}}(t)$ in Eqs. (1). Sensor inputs were generated from the following unrealistically severe integrated angular-rate and specific-force acceleration profiles:

$$\alpha_x = -1.1\tau + 0.9\tau^2 - 0.6\tau^3 + 1.1\tau^4 - 0.1\tau^5 + 0.2\sin 2.6\pi\tau + 0.3(1 - \cos 2.6\pi\tau)\ \text{rad}$$

$$\alpha_y = -0.5\tau + 1.0\tau^2 + 0.3\tau^3 + 0.7\tau^4 + 0.2\tau^5 - 0.5\sin 2.6\pi\tau + 0.6(1 - \cos 2.6\pi\tau)\ \text{rad}$$

$$\alpha_z = 0.3\tau - 1.2\tau^2 + 2\tau^3 - 0.9\tau^4 + 0.4\tau^5 + 0.8\sin 2.6\pi\tau - 0.9(1 - \cos 2.6\pi\tau)\ \text{rad}$$

$$\upsilon_x = 3.5\tau - 2.3\tau^2 + 1.5\tau^3 + 6.1\tau^4 - 2.7\tau^5 - 8\sin 2.6\pi\tau + 5(1 - \cos 2.6\pi\tau)\ \text{ft/s}$$

$$\upsilon_y = 7.3\tau + 1.5\tau^2 - 2.7\tau^3 - 3.6\tau^4 + 1.9\tau^5 + 4\sin 2.6\pi\tau + 3(1 - \cos 2.6\pi\tau)\ \text{ft/s}$$

$$\upsilon_z = -9\tau - 5.6\tau^2 + 4.6\tau^3 + 4.3\tau^4 - 3.5\tau^5 - 7\sin 2.6\pi\tau - 6(1 - \cos 2.6\pi\tau)\ \text{ft/s} \qquad \text{(A1)}$$

where τ is normalized time since t_{m-1} as a fraction of the m cycle time interval, that is, $\tau \equiv (t - t_{m-1})/T_m$. The $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ inputs to the navigation parameter rate equations were calculated as the difference between successive digital integration cycles of Eqs. (A1) divided by the 0.1-μs integration cycle time.

From its definition, τ in Eqs. (A1) runs from 0 to 1 as t runs from t_{m-1} to t_m. Note in Eqs. (A1) that the maximum values for the α and υ sinusoidal terms are of the same order of magnitude over an m cycle, (i.e., from $\tau = 0$ to $\tau = 1$) and the maximum values for the α and υ quadratic expansion terms (at $\tau = 1$, i.e., $t = t_m$) are of the same order of magnitude as the sinusoidal term amplitudes. Thus, from $\tau = 0$ to $\tau = 1$ (i.e., t from t_{m-1} to t_m) all terms for $\boldsymbol{\alpha}$ and for $\boldsymbol{\upsilon}$ are of the same order of magnitude, hence, contribute noticeably to the composite navigation parameter results at $\tau = 1$ (i.e., $t = t_m$). Note also that all sinusoids go through 1.3 cycles (2.6/2) over an m cycle, which is not an integer multiple of the m cycle period.

The exact and reference solutions were generated using an m cycle time period of 0.1 s (10 Hz). After one m cycle, the reference solution [from Eqs. (2)] was as follows (shown rounded):

$$\boldsymbol{\phi}_m = (1.34, 2.16, 1.48)^T\ \text{rad}$$

$$\Delta \boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}} = (-15.7, -2.93, -0.426)^T\ \text{ft/s}$$

$$\Delta \boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}} = (-0.203, -0.133, -1.50)^T\ \text{ft} \qquad \text{(A2)}$$

In Eqs. (A2), $\boldsymbol{\phi}_m$ is the rotation vector equivalent to $C_{B_m}^{B_{m-1}}$ generated by integrating $\dot{C}_{B(t)}^{B_{m-1}}$ in Eqs. (1). The $C_{B_m}^{B_{m-1}}$ to $\boldsymbol{\phi}_m$ conversion formula is given in Ref. 7, Sec. 3.2.2.2.

The maximum magnitudes of the Eqs. (A2) parameters during the 0.1-s integration period and the time when the maximum occurred were as follows (shown rounded):

$$\phi_{\max} = 2.94\ \text{rad at } 0.1\ \text{s} \qquad \Delta\upsilon_{\mathrm{SF}_{\max}} = 24.9\ \text{ft/s at } 0.0559\ \text{s}$$

$$\Delta R_{\mathrm{SF}_{\max}} = 1.52\ \text{ft at } 0.1\ \text{s} \qquad \text{(A3)}$$

The error in the navigation parameters determined with Eqs. (15–17) (calculated by subtraction of the integrated Equations (1) reference results) is shown as a fraction of the Eqs. (A3) magnitudes as

$$\delta\boldsymbol{\phi}_m/\phi_{\max} = (0.750E\text{–}13, 0.153E\text{–}11, -0.257E\text{–}11)^T\ \text{parts}$$

$$\delta\Delta\boldsymbol{v}_{\mathrm{SF}_m}^{B_{m-1}} \Big/ \Delta\upsilon_{\mathrm{SF}_{\max}} = (-0.102E\text{–}11, 0.675E\text{–}13, -0.101E\text{–}11)^T\ \text{parts}$$

$$\delta\Delta\boldsymbol{R}_{\mathrm{SF}_m}^{B_{m-1}} \Big/ \Delta R_{\mathrm{SF}_{\max}} = (0.370E\text{–}11, 0.211E\text{–}12, 0.355E\text{–}12)^T\ \text{parts} \qquad \text{(A4)}$$

The minor error is attributed to computer roundoff and/or trapezoidal integration error. Results clearly indicate that under extreme angle rotation conditions, Eqs. (15–17) continuous-form velocity/position algorithms are alternate exact representations of velocity/position determined by integrating Eqs. (1).

Picard Solution Validation

For this test, the Picard expansion solution for $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ obtained by integrating Eqs. (20) with Eqs. (26–28) was compared with an exact reference solution for $\boldsymbol{\phi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ generated by integrating Eqs. (15) with Eqs. (16). The test included preliminary simulation verification of Eqs. (25) approximations for f under small-angle input conditions. As in the exact solution test, sensor inputs for the Picard test were generated using Eqs. (A1) except that, to reduce the magnitude of the rotation vector $\boldsymbol{\phi}$ for compatibility with Picard expansion accuracy assumptions, the $\boldsymbol{\alpha}$ integrated-rate input components were set to 1/30th of the Eqs. (A1) values. Equations (A1) were used as shown for the $\boldsymbol{\upsilon}$ integrated specific force acceleration components. The $\boldsymbol{\omega}$ and $\boldsymbol{a}_{\mathrm{SF}}$ inputs to the navigation parameter rate equations were calculated as the difference between successive digital integration cycles of Eqs. (A1) divided by the 0.1-μs integration cycle time.

The Picard solution test was initially conducted using an m cycle time period of 0.1 s (10 Hz). After one m cycle, the reference navigation solution [from the integral of Eqs. (15) and (16) previously shown to be exact] was as follows (shown rounded):

$$\boldsymbol{\phi}_m = (0.0264, 0.0690, 0.00674)^T\ \text{rad}$$

$$\boldsymbol{\eta}_m = (5.26, 11.5, -23.9)^T\ \text{ft/s}$$

$$\boldsymbol{\zeta}_m = (0.552, 0.591, -1.15)^T\ \text{ft} \qquad \text{(A5)}$$

The maximum magnitude of the Eqs. (A5) parameters during the 0.1-s integration period and the time when the maximum occurred were as follows (shown rounded):

$$\phi_{\max} = 0.0837\ \text{rad at } 0.0476\ \text{s} \qquad \eta_{\max} = 27.1\ \text{ft/s at } 0.1\ \text{s}$$

$$\zeta_{\max} = 1.40\ \text{ft at } 0.1\ \text{s} \qquad \text{(A6)}$$

The error in the equivalent integrated Picard expansion solution Eqs. (20) with Eqs. (26–28) is shown next as a fraction of the Eqs. (A5) magnitudes,

$$\delta\boldsymbol{\phi}_{\mathrm{Pic}_m} \Big/ \phi_{\max} = (0.294E\text{–}6, -0.262E\text{–}6, -0.172E\text{–}6)^T\ \text{parts}$$

$$\delta\boldsymbol{\eta}_{\mathrm{Pic}_m} \Big/ \eta_{\max} = (-0.199E-5, -0.208E\text{–}5, 0.531E\text{–}5)^T\ \text{parts}$$

$$\delta\boldsymbol{\zeta}_{\mathrm{Pic}_m} \Big/ \zeta_{\max} = (-0.345E\text{–}5, -0.279E\text{–}5, -0.196E\text{–}5)^T\ \text{parts} \qquad \text{(A7)}$$

The Eqs. (A7) results demonstrate that the Picard expansion solution in Eqs. (20) with Eqs. (26–28) is an accurate approximation of the integrated Eqs. (15) and (16) exact solution. For the Picard expansion order in Eqs. (26–28) (fourth order for ϕ, η, and fifth order for ζ), the normalized error magnitudes should be on the order of $\phi_{\max}^4$, which, for $\phi_{\max} = 0.101$ rad in Eqs. (A6), is consistent with Eqs. (A7) results.

To obtain further confidence in the validity of Eqs. (20) with Eqs. (26–28), the preceding test was repeated with a more representative m cycle time period of 0.01 s (100 Hz) using Eqs. (A1) input integrated angular-rate/specific-force profile but with the α components at 1/300th of the Eq. (A1) values and the v components at 1/10th the Eqs. (A1) values, that is, 10% of the earlier 10-Hz test values. The maximum magnitudes of the integrated Eqs. (15) and (16) reference navigation parameters during the 0.01-s integration period and the time when the maximum occurred were as follows (shown rounded):

$$\phi_{\max} = 0.00837 \text{ rad at } 0.00476 \text{ s} \qquad \eta_{\max} = 2.71 \text{ ft/s at } 0.01 \text{ s}$$
$$\zeta_{\max} = 0.0140 \text{ ft at } 0.01 \text{ s} \tag{A8}$$

The equivalent to the Eqs. (A7) results after one 0.01-s m cycle was as follows (normalized by the Eqs. (A8) maximum values):

$$\delta\phi_{\text{Pic}_m}/\phi_{\max} = (0.305E\text{–}10, -0.259E\text{–}10, -0.183E\text{–}10)^T \text{ parts}$$
$$\delta\eta_{\text{Pic}_m}/\eta_{\max} = (-0.159E\text{–}8, -0.232E\text{–}8, 0.506E\text{–}8)^T \text{ parts}$$
$$\delta\zeta_{\text{Pic}_m}/\zeta_{\max} = (-0.314E\text{–}8, -0.306E\text{–}8, -0.212E\text{–}8)^T \text{ parts} \tag{A9}$$

The Eqs. (A9) results clearly indicate that, for a 100-Hz algorithm update rate, the integrated Eqs. (20) with Eqs. (26–28) Picard solution very closely matches the exact integrated Eqs. (15) and (16) reference solution under severe dynamic conditions. Furthermore, Eqs. (A9) were generated using 1/10th the input magnitude as Eqs. (A7) with the η and ζ errors in Eqs. (A9) then becoming $(1/10)^3$ of the errors in Eqs. (A7). This demonstrates that the normalized Picard expansion errors vary by the third power of ϕ, which is equivalent to the unnormalized errors being fourth order in powers of ϕ, η, and ζ, the expected result for the Picard expansion in Eqs. (26–28). Interestingly, the normalized ϕ errors in Eqs. (A9) are $(1/10)^4$ of the errors in Eqs. (A7); hence, the unnormalized ϕ errors are fifth order in powers of ϕ, that is, one order more accurate than what would be expected from the Picard expansion for ϕ in Eq. (26). This unusual result can be traced to the Eqs. (25) approximation of $f_5 \approx 1/12$, whose error is in powers of ϕ^2, not powers of ϕ. The effect on the Eqs. (26) expansion is to eliminate the fourth-order error.

Maneuver-Induced Error Equation Validation

For this test series, the Eqs. (44–46) estimates for ϕ, η, and ζ algorithm error under maneuvers were compared with a reference algorithm error generated directly as the algorithm solution minus the Picard expansion solution obtained by integrating Eqs. (20) with Eqs. (26–28). Note that Eqs. (44–46) were derived as the algorithm solution minus Eqs. (20) [with Eqs. (26–28)]; hence, the Picard solution is the proper reference for analytical derivation accuracy assessment. The tests were conducted using the following angular-rate/specific-force input dynamic maneuver profile:

$$\omega_x = 1.1 + 0.9(t - t_{m-1}) - 0.6\left[(t - t_{m-1})^2/2!\right] + 1.1\left[(t - t_{m-1})^3/3!\right] - 0.1\left[(t - t_{m-1})^4/4!\right] \text{ rad/s}$$
$$\omega_y = -0.5 + 1.0(t - t_{m-1}) + 0.3\left[(t - t_{m-1})^2/2!\right] + 0.7\left[(t - t_{m-1})^3/3!\right] + 0.2\left[(t - t_{m-1})^4/4!\right] \text{ rad/s}$$
$$\omega_z = 0.3 - 1.2(t - t_{m-1}) + 2\left[(t - t_{m-1})^2/2!\right] - 0.9\left[(t - t_{m-1})^3/3!\right] + 0.4\left[(t - t_{m-1})^4/4!\right] \text{ rad/s}$$

$$a_{\text{SF}_x} = 3.5 - 2.3(t - t_{m-1}) + 1.5\left[(t - t_{m-1})^2/2!\right] + 6.1\left[(t - t_{m-1})^3/3!\right] - 2.7\left[(t - t_{m-1})^4/4!\right] \text{ ft/s}^2$$
$$a_{\text{SF}_y} = 7.3 + 1.5(t - t_{m-1}) - 2.7\left[(t - t_{m-1})^2/2!\right] - 3.6\left[(t - t_{m-1})^3/3!\right] + 1.9\left[(t - t_{m-1})^4/4!\right] \text{ ft/s}^2$$
$$a_{\text{SF}_z} = -9 - 5.6(t - t_{m-1}) + 4.6\left[(t - t_{m-1})^2/2!\right] + 4.3\left[(t - t_{m-1})^3/3!\right] - 3.5\left[(t - t_{m-1})^4/4!\right] \text{ ft/s}^2 \tag{A10}$$

Note in Eqs. (A10) compared to the generic equations (39) form [and the input to Eqs. (44–46)] that the coefficients in Eqs. (A10) multiplying the $(t - t_{m-1})^k/k!$ terms represent the kth derivative of the angular-rate/specific-force components. They were selected to be of the same order of magnitude to assure comparable participation in Eqs. (44–46) and its reference equivalent.

Using Eqs. (A10) and a 0.1-s integration time, the exact integrated Eqs. (15) solution parameters and their maximum values at 0.1 s were as follows:

$$\phi = (0.114, -0.0448, 0.0245)^T \text{ rad}$$
$$\eta = (0.339, 0.737, -0.927)^T \text{ ft/s}$$
$$\zeta = (0.0171, 0.0367, -0.0459)^T \text{ ft}$$
$$\phi_{\max} = 0.125 \text{ rad at } 0.1 \text{ s} \qquad \eta_{\max} = 1.23 \text{ ft/s at } 0.1 \text{ s}$$
$$\zeta_{\max} = 0.0613 \text{ ft at } 0.1 \text{ s} \tag{A11}$$

The algorithm errors under the Eqs. (A10) input were calculated at 0.1 s as the algorithm solution minus the integrated equations (20) with Eqs. (26–28) Picard solution. As an example, for ϕ and η algorithm c, Eqs. (44) and (45) error vector component and magnitude results at 0.1 s were as follows:

$$\delta\phi_{\text{AlgcEr}} = (0.121E\text{–}6, -0.944E\text{–}7, 0.528E\text{–}7)^T \text{ rad}$$
$$\delta\eta_{\text{AlgcEr}} = (0.598E\text{–}6, -0.801E\text{–}7, -0.117E\text{–}6)^T \text{ ft/s}$$
$$\delta\phi_{\text{AlgcEr}_{\text{Mag}}} = 0.162E\text{–}6 \text{ rad}$$
$$\delta\eta_{\text{AlgcEr}_{\text{Mag}}} = 0.614E\text{–}6 \text{ ft/s} \tag{A12}$$

The Eqs. (44) and (45) analytical predictions of the ϕ and η algorithm c errors were then calculated and compared to the Eqs. (A12) reference model equivalent. The results were as follows [normalized by the Eqs. (A12) magnitude values]:

$$\delta(\delta\phi_{\text{AnalAlgcEr}})/\delta\phi_{\text{AlgcEr}_{\text{Mag}}} = (0.0442, -0.0490, 0.0304)^T \text{ parts}$$
$$\delta(\delta\eta_{\text{AnalAlgcEr}})/\delta\eta_{\text{AlgcEr}_{\text{Mag}}} = (0.159, 0.0823, 0.0811)^T \text{ parts} \tag{A13}$$

Equations (A13) results demonstrate that for the 0.1-s integration time (equivalent to m cycle sampling at a 10-Hz cycle rate) Eqs. (44) provide a good analytical estimate of the ϕ and η algorithm c errors.

The preceding test was repeated using the Eqs. (A10) input with a 0.01-s integration time. Results were as follows:

$$\phi_{\max} = 0.0125 \text{ rad} \qquad \eta_{\max} = 0.121 \text{ ft/s} \qquad \zeta_{\max} = 0.000606 \text{ ft}$$
$$\delta\phi_{\text{AlgcEr}_{\text{Mag}}} = 0.172E\text{–}11 \text{ rad} \qquad \delta\eta_{\text{AlgcEr}_{\text{Mag}}} = 0.690E\text{–}11 \text{ ft/s}$$
$$\delta(\delta\phi_{\text{AnalAlgcEr}})/\delta\phi_{\text{AlgcEr}_{\text{Mag}}} = (0.00441, -0.00483, 0.00291)^T \text{ parts}$$
$$\delta(\delta\eta_{\text{AnalAlgcEr}})/\delta\eta_{\text{AlgcEr}_{\text{Mag}}} = (0.0144, 0.00802, 0.00743)^T \text{ parts} \tag{A14}$$

Equations (A14) show that for the 0.01-s integration time (equivalent to a 100-Hz m cycle rate) Eqs. (44) and (45) provide a factor of 10 improved accuracy in estimating ϕ and η algorithm c error

compared to the 0.1-s m cycle result. This is consistent with what is expected when the m cycle rate is increased by a factor of 10 for the same angular-rate/specific-force input.

The preceding results are typical for Eqs. (44–46), providing further confidence in the validity of these equations for estimating algorithm error.

Vibration-Induced Error Equation Validation

Algorithm error equations (49–51) were validated using the Eqs. (47) vibration profile with $\theta_0 = 0.001$ rad, $a_{SF_0} = 3\ g$, and Ω spanning a range of vibration frequencies less and greater than the m cycle update frequency. Two sets of tests were conducted. For the first, Eqs. (49–51) analytical predictions were compared to the equivalent results generated by direct integration of the algorithm rate equations minus the equivalent integrated equations (15) exact solution (already verified). The comparison results were similar to those for the Picard accuracy test discussed earlier, providing confidence in the validity of Eqs. (49–51). For the second test series, Eqs. (49–51) analytical predictions were compared with the integrated Eqs. (36–38) equivalent results. Because Eqs. (49–51) were derived from Eqs. (36–38) without approximations, the second test was expected to provide an exact match of Eqs. (49–51) with integrated equations (36–38). It did within computer roundoff error (similar to the exact solution test result discussed earlier), providing further confidence in Eqs. (49–51).

Acknowledgment

The author wishes to express his appreciation to Kelly M. Roscoe of Applied Strapdown Analytics for her assistance in the analytical verification of equation derivation accuracy and for providing useful comments to improve manuscript comprehension.

References

[1]Savage, P. G., "A New Second-Order Solution for Strapped-Down Attitude Computation," AIAA/JACC Guidance and Control Conf., Aug. 1966.

[2]Jordan, J. W., "An Accurate Strapdown Direction Cosine Algorithm," NASA TN-D-5384, Sept. 1969.

[3]Goodman, L. E., and Robinson, A. R., "Effects of Finite Rotations on Gyroscope Sensing Devices," *Journal of Applied Mechanics*, Vol. 25, June 1958, pp. 210–213.

[4]Bortz, J. E., "A New Mathematical Formulation for Strapdown Inertial Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-7, No. 1, 1971, pp. 61–66.

[5]Laning, J. H., Jr., "The Vector Analysis of Finite Rotations and Angles," Massachusetts Instrumentation Lab. Special Rept. 6398-S-3, Massachusetts Inst. of Technology, Cambridge, MA, Sept. 1949.

[6]Savage, P. G., "Strapdown Inertial Navigation System Integration Algorithm Design, Part 2: Velocity and Position Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 208–221.

[7]Savage, P. G., *Strapdown Analytics*, Strapdown Associates, Inc., Maple Plain, MN, 2000.

[8]Savage, P. G., "Strapdown System Algorithms," *Advances in Strapdown Inertial Systems*, NATO AGARD Lecture Ser. No. 133, May 1984, Sec. 3.

[9]Martin, W. T., and Reissner, E., *Elementary Differential Equations*, Addison–Wesley, Cambridge, MA, 1956, Sec. 6-8.

[10]Savage, P. G., "Strapdown Inertial Navigation System Integration Algorithm Design, Part 1: Attitude Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 19–28.

[11]Shuster, M. D., "The Kinematic Equation for the Rotation Vector," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 1, 1993, pp. 263–267.

[12]Miller, R., "A New Strapdown Attitude Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 4, 1983, pp. 287–291.

[13]Ignagni, M. B., "Optimal Strapdown Attitude Integration Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 363–369.

[14]Ignagni, M. B., "On the Orientation Vector Differential Equation in Strapdown Inertial Systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 30, No. 4, 1994, pp. 1076–1081.

[15]Roscoe, K. M., "Equivalency Between Strapdown Inertial Navigation Coning and Sculling Integrals/Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 201–205.