

# Equivalency Between Strapdown Inertial Navigation Coning and Sculling Integrals/Algorithms

Kelly M. Roscoe\*

KMR Associates, Orono, Minnesota 55356

**This paper develops a generic equivalency between strapdown inertial navigation coning and sculling integrals and algorithms. The equivalency allows a previously derived coning algorithm to be converted to its corresponding sculling algorithm using a simple mathematical formula. Two examples are provided illustrating the coning-to-sculling algorithm conversion process. The results are verified by comparing them against previously derived coning and sculling algorithms.**

## Introduction

TWO key calculations performed in strapdown inertial navigation systems are updating the body frame (inertial sensor axes) attitude and updating the vehicle velocity. The attitude update calculation includes an integral term (denoted as *coning*) that is nonzero when the vehicle's angular rate vector is rotating. The velocity update calculation includes an integral term (denoted as *sculling*) that is nonzero when the vehicle's angular rate or specific force acceleration vector is rotating, or when the ratio of the angular rate to specific force acceleration magnitude is not constant.

To improve the accuracy of the attitude and velocity update calculations, particularly in environments where the angular rate vector or specific force acceleration vector rotation rate is large, high-rate algorithms have been developed for the coning and sculling integrals. The first detailed optimization of algorithms for the coning integral appeared in a paper by R. Miller.<sup>1</sup> Miller's procedure was then applied and extended in a variety of papers, two of which are Refs. 2 and 3. A detailed description of the coning integral algorithm design process is provided in Ref. 4. Work on the design of sculling algorithms has not been as extensive as that of coning algorithms. Some recent work detailing the design of sculling algorithms is provided in Refs. 5–7. In Ref. 5 Savage provides an analytical description of sculling in two forms; the first having only one term, denoted as the *composite sculling/velocity rotation compensation integral*, and the second having two terms, denoted as *velocity rotation compensation and the sculling integral*. An example is provided that develops a digital algorithm for calculating the sculling integral term. In Ref. 6 Ignagni derives a class of optimized sculling algorithms for the composite sculling/velocity rotation compensation integral and demonstrates a duality between the derived class of sculling algorithms and a previously derived class of coning algorithms. In addition, Ref. 6 provides a detailed example illustrating the derivation of one sculling algorithm solution and compares it to a previously derived coning algorithm solution (Ref. 2, algorithm 3). In Ref. 7 Mark and Tazartes develop a sculling algorithm using a different approach than that in Refs. 5 and 6. Both approaches are valid and have been successfully applied in strapdown systems. This paper only deals with sculling algorithm forms found in Refs. 5 and 6.

This paper develops a generic equivalency between the coning and sculling integrals and extends it to algorithms that take the same form as those in Refs. 5 and 6. The equivalency allows one to convert an already derived coning algorithm to its sculling algorithm counterpart using a simple mathematical formula. The paper first introduces the coning and sculling integral equations. Generic integral/algorithm equivalencies are then developed showing how con-

ing algorithms can be converted to their sculling algorithm equivalents. Two examples of the coning-to-sculling conversion process are then provided and compared with results that were derived earlier. Example 1 converts the coning algorithm developed in Ref. 4 to its sculling algorithm counterpart (developed in Ref. 5). Example 2 converts the coning algorithm in Ref. 2 (algorithm 3) (derived in Ref. 3, algorithm F) to its sculling algorithm counterpart (derived in Ref. 6, algorithm 2).

## Coning and Sculling Integrals

In strapdown inertial systems updating of the attitude direction cosine matrix (or quaternion) includes solving for the rotation vector that defines the body attitude (inertial sensors' orientation) at time  $t_m$  relative to the body attitude at time  $t_{m-1}$  (one computer cycle earlier). The general rate equation for this rotation vector (proposed for use in strapdown inertial navigation by Bortz<sup>8</sup>) is

$$\dot{\phi} = \omega + \frac{1}{2}\phi \times \omega + \frac{1}{\phi^2} \left[ 1 - \frac{\phi \sin \phi}{2(1 - \cos \phi)} \right] \phi \times (\phi \times \omega) \quad (1)$$

where  $\phi$  is the rotation vector defining the body attitude at some time  $t$  (greater than  $t_{m-1}$ ) relative to the body attitude at time  $t_{m-1}$ ,  $\phi$  is the magnitude of  $\phi$ , and  $\omega$  is the angular rate vector measured by strapdown angular rate sensors (expressed with coordinates in the body frame). To second-order accuracy, as discussed in Ref. 4, Eqs. (29–33), Eq. (1) reduces to

$$\dot{\phi} \approx \omega + \frac{1}{2}\alpha(t) \times \omega \quad (2)$$

where

$$\alpha(t) = \int_{t_{m-1}}^t \omega \, d\tau$$

From Eq. (2) we can write the equation for the rotation vector that defines the body attitude at time  $t_m$  relative to the body attitude at time  $t_{m-1}$  as

$$\phi(m) = \int_{t_{m-1}}^{t_m} \left\{ \omega + \frac{1}{2}[\alpha(t) \times \omega] \right\} dt \quad (3)$$

where  $\phi(m)$  is the rotation vector defining the body attitude at time  $t_m$  relative to the body attitude at time  $t_{m-1}$ . The second term in Eq. (3), shown separately next, has been designated the coning integral  $\theta_c$  and is nonzero when the angular rate vector  $\omega$  is rotating:

$$\theta_c(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2}[\alpha(t) \times \omega] dt \quad (4)$$

A substantial number of digital integration algorithms have been designed for the coning integral  $[\theta_c(m)]$  to improve the attitude accuracy in strapdown systems without sacrificing computer throughput. Examples of these algorithms can be found in Refs. 2–4.

Received 25 May 2000; revision received 11 October 2000; accepted for publication 1 November 2000. Copyright © 2000 by Kelly M. Roscoe. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

\*Owner, 4030 Elm Street. Member AIAA.

The velocity update calculation includes solving for an integral that represents the change in velocity (in body frame coordinates) from time  $t_{m-1}$  to time  $t_m$  caused by specific force acceleration [see Ref. 5, Eq. (26)]:

$$\Delta \mathbf{v}(m) = \mathbf{v}(m) + \int_{t_{m-1}}^{t_m} [\boldsymbol{\alpha}(t) \times \mathbf{a}] dt \quad (5)$$

where

$$\boldsymbol{\alpha}(t) = \int_{t_{m-1}}^t \boldsymbol{\omega} d\tau, \quad \mathbf{v}(m) = \int_{t_{m-1}}^{t_m} \mathbf{a} dt$$

and where  $\Delta \mathbf{v}(m)$  is the change in velocity caused by specific force acceleration from time  $t_{m-1}$  to time  $t_m$  (expressed with coordinates in the body frame) and  $\mathbf{a}$  is the nongravitational acceleration vector measured by strapdown accelerometers (expressed with coordinates in the body frame). Equivalently, Eq. (5) can be written as [see Ref. 5, Eqs. (27–36) for development]

$$\begin{aligned} \Delta \mathbf{v}(m) = & \mathbf{v}(m) + \frac{1}{2} [\boldsymbol{\alpha}(m) \times \mathbf{v}(m)] \\ & + \int_{t_{m-1}}^{t_m} \frac{1}{2} [\boldsymbol{\alpha}(t) \times \mathbf{a} + \mathbf{v}(t) \times \boldsymbol{\omega}] dt \end{aligned} \quad (6)$$

where

$$\boldsymbol{\alpha}(m) = \int_{t_{m-1}}^{t_m} \boldsymbol{\omega} d\tau, \quad \mathbf{v}(t) = \int_{t_{m-1}}^t \mathbf{a} d\tau$$

The second and third terms in Eq. (6), shown separately next, have been designated as velocity rotation compensation ( $\Delta \mathbf{v}_{\text{rot}}$ ) and sculling ( $\Delta \mathbf{v}_{\text{scul}}$ ), respectively:

$$\Delta \mathbf{v}_{\text{rot}}(m) = \frac{1}{2} [\boldsymbol{\alpha}(m) \times \mathbf{v}(m)] \quad (7)$$

$$\Delta \mathbf{v}_{\text{scul}}(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2} [\boldsymbol{\alpha}(t) \times \mathbf{a} + \mathbf{v}(t) \times \boldsymbol{\omega}] dt \quad (8)$$

Comparing Eqs. (6–8) with Eq. (5) shows

$$\int_{t_{m-1}}^{t_m} [\boldsymbol{\alpha}(t) \times \mathbf{a}] dt = \Delta \mathbf{v}_{\text{rot}}(m) + \Delta \mathbf{v}_{\text{scul}}(m) \quad (9)$$

from which we define

$$\Delta \mathbf{v}_{\text{rot/scul}}(m) \equiv \Delta \mathbf{v}_{\text{rot}}(m) + \Delta \mathbf{v}_{\text{scul}}(m) = \int_{t_{m-1}}^{t_m} [\boldsymbol{\alpha}(t) \times \mathbf{a}] dt \quad (10)$$

where  $\Delta \mathbf{v}_{\text{rot/scul}}(m)$  is the composite sculling and velocity rotation compensation term.

As with the coning integral, algorithms for the sculling integral [ $\Delta \mathbf{v}_{\text{scul}}(m)$ ] have been designed to improve the accuracy of the strapdown system velocity. Examples of such algorithms are provided in Refs. 5 and 6. In Ref. 5 a digital algorithm for calculating  $\Delta \mathbf{v}_{\text{scul}}(m)$  is provided, and in Ref. 6 a class of optimized algorithms for calculating  $\Delta \mathbf{v}_{\text{rot/scul}}(m)$  is derived. By comparing the derived class of sculling algorithms with an already derived class of coning algorithms, Ref. 6 demonstrates a duality between the two. An example illustrating this duality is provided, comparing one sculling algorithm to its coning algorithm counterpart (Ref. 2, algorithm 3).

The remainder of this paper shows the development of a simple mathematical formula that converts already derived coning algorithms to their sculling algorithm equivalents, utilizing generic equivalencies that exist between the coning and sculling integrals.

## Generic Integral/Algorithm Equivalencies

Let us define a vector  $\mathbf{U}_1$  to be the integral of the cross product of two vectors  $\mathbf{V}_1$  and  $\mathbf{v}_1$ :

$$\mathbf{U}_1 \equiv \int (\mathbf{V}_1 \times \mathbf{v}_1) dt \quad (11)$$

where

$$\mathbf{V}_1 = \int \mathbf{v}_1 dt \quad (12)$$

and  $\mathbf{v}_1$  is an arbitrary vector. Let us also define  $\hat{\mathbf{U}}_1$  to be a digital integration algorithm for  $\mathbf{U}_1$ . Similarly, let  $\mathbf{U}_2$  be the integral of  $\mathbf{V}_2 \times \mathbf{v}_2$ , where  $\mathbf{v}_2$  is another arbitrary vector:

$$\mathbf{U}_2 \equiv \int (\mathbf{V}_2 \times \mathbf{v}_2) dt \quad (13)$$

where

$$\mathbf{V}_2 = \int \mathbf{v}_2 dt \quad (14)$$

and  $\hat{\mathbf{U}}_2$  is a digital integration algorithm for  $\mathbf{U}_2$ . Because  $\mathbf{U}_1$  and  $\mathbf{U}_2$  have identical mathematical forms,  $\mathbf{U}_2$  equals  $\mathbf{U}_1$  when  $\mathbf{v}_1$  in  $\mathbf{U}_1$  is replaced by  $\mathbf{v}_2$ . Similarly,  $\hat{\mathbf{U}}_2$  equals  $\hat{\mathbf{U}}_1$  when  $\mathbf{v}_1$  terms in  $\hat{\mathbf{U}}_1$  are replaced by  $\mathbf{v}_2$  terms. Summing the integrals and extending to more terms finds

$$\begin{aligned} \mathbf{U}_1 \pm \mathbf{U}_2 \pm \mathbf{U}_3 \pm \dots = & \mathbf{U}_1 \pm (\mathbf{U}_1 \text{ with } \mathbf{v}_1 \text{ replaced by } \mathbf{v}_2) \\ & \pm (\mathbf{U}_1 \text{ with } \mathbf{v}_1 \text{ replaced by } \mathbf{v}_3) \pm \dots \end{aligned} \quad (15)$$

and it follows that for the digital integration algorithms

$$\begin{aligned} \hat{\mathbf{U}}_1 \pm \hat{\mathbf{U}}_2 \pm \hat{\mathbf{U}}_3 \pm \dots = & \hat{\mathbf{U}}_1 \\ & \pm (\hat{\mathbf{U}}_1 \text{ with } \mathbf{v}_1 \text{ terms replaced by } \mathbf{v}_2 \text{ terms}) \\ & \pm (\hat{\mathbf{U}}_1 \text{ with } \mathbf{v}_1 \text{ terms replaced by } \mathbf{v}_3 \text{ terms}) \pm \dots \end{aligned} \quad (16)$$

Now define the following:

$$\mathbf{v}_3 \equiv \mathbf{v}_1 + \mathbf{v}_2 \quad (17)$$

$$\mathbf{U}_3 \equiv \int (\mathbf{V}_3 \times \mathbf{v}_3) dt \quad (18)$$

where

$$\mathbf{V}_3 = \int \mathbf{v}_3 dt \quad (19)$$

and  $\hat{\mathbf{U}}_3$  is a digital integration algorithm for  $\mathbf{U}_3$ . From the discussion following Eq. (14),  $\hat{\mathbf{U}}_3$  equals  $\hat{\mathbf{U}}_1$  when  $\mathbf{v}_1$  terms in  $\hat{\mathbf{U}}_1$  are replaced by  $\mathbf{v}_3$  terms.

Substituting Eqs. (17), (12), and (14) in (19) gives

$$\mathbf{V}_3 = \mathbf{V}_1 + \mathbf{V}_2 \quad (20)$$

Taking the cross product of Eq. (20) and (17) obtains

$$\begin{aligned} \mathbf{V}_3 \times \mathbf{v}_3 = & (\mathbf{V}_1 + \mathbf{V}_2) \times (\mathbf{v}_1 + \mathbf{v}_2) \\ = & (\mathbf{V}_1 \times \mathbf{v}_1) + (\mathbf{V}_1 \times \mathbf{v}_2) \\ & + (\mathbf{V}_2 \times \mathbf{v}_1) + (\mathbf{V}_2 \times \mathbf{v}_2) \end{aligned} \quad (21)$$

Integrating Eq. (21), substituting Equations (11), (13), and (18), and rearranging yields

$$\int [(\mathbf{V}_1 \times \mathbf{v}_2) + (\mathbf{V}_2 \times \mathbf{v}_1)] dt = \mathbf{U}_3 - \mathbf{U}_1 - \mathbf{U}_2 \quad (22)$$

Let  $\mathbf{U}_4$  equal the left side of Eq. (22):

$$\mathbf{U}_4 = \int [(\mathbf{V}_1 \times \mathbf{v}_2) + (\mathbf{V}_2 \times \mathbf{v}_1)] dt \quad (23)$$

and let  $\hat{U}_4$  be a digital integration algorithm for  $U_4$ . Substituting Eq. (23) in Eq. (22) gives

$$U_4 = U_3 - U_1 - U_2 \quad (24)$$

With the definition of  $\hat{U}_4$  and the generic digital integration algorithm equivalency developed earlier [Eq. (16)], it follows from Eq. (24) that

$$\begin{aligned} \hat{U}_4 &= \hat{U}_3 - \hat{U}_1 - \hat{U}_2 \\ &= (\hat{U}_1 \text{ with } v_1 \text{ terms replaced by } v_3 \text{ terms}) - \hat{U}_1 \\ &\quad - (\hat{U}_1 \text{ with } v_1 \text{ terms replaced by } v_2 \text{ terms}) \end{aligned} \quad (25)$$

Equation (25) is a simple mathematical formula that converts a digital integration algorithm for  $U_1$  to a digital integration algorithm for  $U_4$ . Therefore, if  $\hat{U}_1$  has been derived, using Eq. (25) to obtain  $\hat{U}_4$  eliminates a separate, potentially lengthy,  $\hat{U}_4$  derivation.

### Applying Generic Equivalencies to Coning/Sculling Integrals

Repeating the coning and sculling integrals [Eqs. (4) and (8)]

$$\theta_c(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2} [\alpha(t) \times \omega] dt, \quad \alpha(t) = \int_{t_{m-1}}^t \omega d\tau \quad (26)$$

$$\Delta v_{\text{scul}}(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2} [\alpha(t) \times a + v(t) \times \omega] dt \quad (27)$$

$$\alpha(t) = \int_{t_{m-1}}^t \omega d\tau, \quad v(t) = \int_{t_{m-1}}^t a d\tau$$

Let us also define the following two integrals:

$$U_{v \times a}(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2} [v(t) \times a] dt \quad (28)$$

$$U_{\alpha v \times \omega a}(m) = \int_{t_{m-1}}^{t_m} \frac{1}{2} [\{\alpha(t) + v(t)\} \times (\omega + a)] dt \quad (29)$$

Comparing Eqs. (26–29) with Eqs. (11–14), (17–19), and (23) obtains the following equivalencies:

$$\begin{aligned} \theta_c &= \frac{1}{2} U_1, & \alpha(t) &= V_1, & \omega &= v_1, & U_{v \times a} &= \frac{1}{2} U_2 \\ v(t) &= V_2, & a &= v_2, & U_{\alpha v \times \omega a} &= \frac{1}{2} U_3 \\ \alpha(t) + v(t) &= V_3, & \omega + a &= v_3, & \Delta v_{\text{scul}} &= \frac{1}{2} U_4 \end{aligned} \quad (30)$$

Substituting the digital integration algorithm equivalents for the Eq. (30)  $\theta_c$ ,  $U_{v \times a}$ ,  $U_{\alpha v \times \omega a}$ ,  $\Delta v_{\text{scul}}$  terms (i.e.,  $\theta_c$ ,  $\hat{U}_{v \times a}$ ,  $\hat{U}_{\alpha v \times \omega a}$ ,  $\Delta \hat{v}_{\text{scul}}$ ) in Eq. (25) gives

$$\begin{aligned} \Delta \hat{v}_{\text{scul}}(m) &= \hat{U}_{\alpha v \times \omega a}(m) - \hat{\theta}_c(m) - \hat{U}_{v \times a}(m) \\ &= \{\hat{\theta}_c(m) \text{ with } \omega \text{ replaced by } (\omega + a) \\ &\quad \text{and } \alpha(t) \text{ replaced by } [\alpha(t) + v(t)]\} - \hat{\theta}_c(m) \\ &\quad - [\hat{\theta}_c(m) \text{ with } \omega \text{ replaced by } a \text{ and } \alpha(t) \text{ replaced by } v(t)] \end{aligned} \quad (31)$$

Equation (31) is a simple mathematical formula that converts a coning algorithm  $[\hat{\theta}_c(m)]$  to its sculling algorithm equivalent  $[\Delta \hat{v}_{\text{scul}}(m)]$ . Therefore, if  $\hat{\theta}_c(m)$  has been derived, Eq. (31) can be used to obtain  $\Delta \hat{v}_{\text{scul}}(m)$  instead of performing a separate and sometimes difficult derivation.

### Examples of the Coning-to-Sculling Algorithm Conversion Process

This section converts two already derived coning algorithms to their sculling algorithm counterparts using Eq. (31). The results are then compared with the separately derived sculling algorithms.

#### Example 1—Coning Algorithm<sup>4</sup>

The coning algorithm derived in Ref. 4 [Eq. (47)] is

$$\hat{\theta}_c(m) = \sum_{l=1}^L \frac{1}{2} \left( \alpha_{l-1} + \frac{1}{6} \Delta \alpha_{l-1} \right) \times \Delta \alpha_l \quad (32)$$

where  $l$  is the computer cycle index within a computer  $m$  cycle, defined to be zero at time  $t_{m-1}$ ;  $L$  is the number of  $l$  cycles within an  $m$  cycle;  $\alpha_{l-1}$  is  $\alpha(t)$  in Eq. (26) at time  $l-1$  within an  $m$  cycle (integral limits from  $m-1$  to  $l-1$ ); and  $\Delta \alpha_l$ ,  $\Delta \alpha_{l-1}$  is  $\alpha(t)$  in Eq. (26) with integral limits from  $l-1$  to  $l$  and from  $l-2$  to  $l-1$ . For  $l=0$ ,  $\Delta \alpha_{l-1}$  is defined as the integral of  $\omega$  over the  $l$  cycle that ends at time  $t_{m-1}$ .

Substituting Eq. (32) in Eq. (31) yields

$$\begin{aligned} \Delta \hat{v}_{\text{scul}}(m) &= [\hat{\theta}_c(m) \text{ with } \alpha_{l-1} \text{ replaced by } (\alpha_{l-1} + v_{l-1}) \\ &\quad \text{and } \Delta \alpha_{l-1}, \Delta \alpha_l \text{ replaced by } (\Delta \alpha_{l-1} + \Delta v_{l-1}), (\Delta \alpha_l \\ &\quad + \Delta v_l)] - \hat{\theta}_c(m) - [\hat{\theta}_c(m) \text{ with } \alpha_{l-1} \text{ replaced by } v_{l-1} \\ &\quad \text{and } \Delta \alpha_{l-1}, \Delta \alpha_l \text{ replaced by } \Delta v_{l-1}, \Delta v_l] \\ &= \sum_{l=1}^L \frac{1}{2} \left( \alpha_{l-1} + v_{l-1} + \frac{1}{6} \Delta \alpha_{l-1} + \frac{1}{6} \Delta v_{l-1} \right) \\ &\quad \times (\Delta \alpha_l + \Delta v_l) - \sum_{l=1}^L \frac{1}{2} \left( \alpha_{l-1} + \frac{1}{6} \Delta \alpha_{l-1} \right) \times \Delta \alpha_l \\ &\quad - \sum_{l=1}^L \frac{1}{2} \left( v_{l-1} + \frac{1}{6} \Delta v_{l-1} \right) \times \Delta v_l \\ &= \sum_{l=1}^L \frac{1}{2} \left( \alpha_{l-1} + \frac{1}{6} \Delta \alpha_{l-1} \right) \times \Delta v_l \\ &\quad + \sum_{l=1}^L \frac{1}{2} \left( v_{l-1} + \frac{1}{6} \Delta v_{l-1} \right) \times \Delta \alpha_l \end{aligned} \quad (33)$$

where  $v_{l-1}$  is  $v(t)$  in Eq. (27) at time  $l-1$  within an  $m$  cycle (integral limits from  $m-1$  to  $l-1$ ) and  $\Delta v_l$ ,  $\Delta v_{l-1}$  is  $v(t)$  in Eq. (27) with integral limits from  $l-1$  to  $l$  and from  $l-2$  to  $l-1$ . For  $l=0$ ,  $\Delta v_{l-1}$  is defined as the integral of  $a$  over the  $l$  cycle that ends at time  $t_{m-1}$ .

The sculling algorithm counterpart to the Ref. 4 coning algorithm is derived [from Eq. (27)] in Ref. 5 [Equation (61)] and is identical to Eq. (33).

#### Example 2—Coning Algorithm 3 (Ref. 2)

Coning algorithm number 3 in Ref. 2 [Eqs. (2) and (3) with algorithm 3 for Eq. (3)] (derived in Ref. 3, algorithm F) is

$$\begin{aligned} \hat{\theta}_c(m) &= \frac{1}{2} \sum_{l=2}^L \alpha_{l-1} \times \Delta \alpha_l \\ &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \alpha_l(1) + \frac{27}{20} \Delta \alpha_l(2) \right] \times \Delta \alpha_l(3) \end{aligned} \quad (34)$$

where  $L$  is the number of  $l$  cycles within an  $m$  cycle [Note that  $m$ ,  $l$ , and  $L$  in Eq. (34) are equivalent, respectively, to  $n$ ,  $m$ , and  $M$  in Ref. 2.];  $\Delta \alpha_l(i)$  is the incremental angle vector over the  $i$ th interval within the  $l$  cycle [The sum of  $\Delta \alpha_l(1)$ ,  $\Delta \alpha_l(2)$ , and  $\Delta \alpha_l(3)$  equals  $\alpha(t)$  in Eq. (26) over an  $l$  cycle (integral limits from  $l-1$  to  $l$ ); and  $\Delta \alpha_l$ ,  $\alpha_{l-1}$  is identical to the Eq. (32) definitions.

Substituting Eq. (34) in Eq. (31) obtains

$$\begin{aligned}
 \Delta \hat{\mathbf{v}}_{\text{scul}}(m) &= \{\hat{\boldsymbol{\theta}}_c(m) \text{ with } \Delta \boldsymbol{\alpha}_l(1), \Delta \boldsymbol{\alpha}_l(2), \Delta \boldsymbol{\alpha}_l(3), \boldsymbol{\alpha}_{l-1}, \Delta \boldsymbol{\alpha}_l \\
 &\quad \text{replaced by } [\Delta \boldsymbol{\alpha}_l(1) + \Delta \mathbf{v}_l(1)], [\Delta \boldsymbol{\alpha}_l(2) + \Delta \mathbf{v}_l(2)], \\
 &\quad [\Delta \boldsymbol{\alpha}_l(3) + \Delta \mathbf{v}_l(3)], (\boldsymbol{\alpha}_{l-1} + \mathbf{v}_{l-1}), (\Delta \boldsymbol{\alpha}_l + \Delta \mathbf{v}_l)\} \\
 &\quad - \hat{\boldsymbol{\theta}}_c(m) - [\hat{\boldsymbol{\theta}}_c(m) \text{ with } \Delta \boldsymbol{\alpha}_l(1), \Delta \boldsymbol{\alpha}_l(2), \Delta \boldsymbol{\alpha}_l(3), \\
 &\quad \boldsymbol{\alpha}_{l-1}, \Delta \boldsymbol{\alpha}_l \text{ replaced by } \Delta \mathbf{v}_l(1), \Delta \mathbf{v}_l(2), \Delta \mathbf{v}_l(3), \mathbf{v}_{l-1}, \Delta \mathbf{v}_l] \\
 &= \frac{1}{2} \sum_{l=2}^L [(\boldsymbol{\alpha}_{l-1} + \mathbf{v}_{l-1}) \times (\Delta \boldsymbol{\alpha}_l + \Delta \mathbf{v}_l)] \\
 &\quad + \sum_{l=1}^L \left\{ \frac{9}{20} [\Delta \boldsymbol{\alpha}_l(1) + \Delta \mathbf{v}_l(1)] + \frac{27}{20} [\Delta \boldsymbol{\alpha}_l(2) + \Delta \mathbf{v}_l(2)] \right\} \\
 &\quad \times [\Delta \boldsymbol{\alpha}_l(3) + \Delta \mathbf{v}_l(3)] - \frac{1}{2} \sum_{l=2}^L (\boldsymbol{\alpha}_{l-1} \times \Delta \boldsymbol{\alpha}_l) \\
 &\quad - \sum_{l=1}^L \left[ \frac{9}{20} \Delta \boldsymbol{\alpha}_l(1) + \frac{27}{20} \Delta \boldsymbol{\alpha}_l(2) \right] \times \Delta \boldsymbol{\alpha}_l(3) \\
 &\quad - \frac{1}{2} \sum_{l=2}^L (\mathbf{v}_{l-1} \times \Delta \mathbf{v}_l) - \sum_{l=1}^L \left[ \frac{9}{20} \Delta \mathbf{v}_l(1) \right. \\
 &\quad \left. + \frac{27}{20} \Delta \mathbf{v}_l(2) \right] \times \Delta \mathbf{v}_l(3) \quad (35)
 \end{aligned}$$

or, after combining terms

$$\begin{aligned}
 \Delta \hat{\mathbf{v}}_{\text{scul}}(m) &= \frac{1}{2} \sum_{l=2}^L [(\boldsymbol{\alpha}_{l-1} \times \Delta \mathbf{v}_l) + (\mathbf{v}_{l-1} \times \Delta \boldsymbol{\alpha}_l)] \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \boldsymbol{\alpha}_l(1) + \frac{27}{20} \Delta \boldsymbol{\alpha}_l(2) \right] \times \Delta \mathbf{v}_l(3) \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \mathbf{v}_l(1) + \frac{27}{20} \Delta \mathbf{v}_l(2) \right] \times \Delta \boldsymbol{\alpha}_l(3) \quad (36)
 \end{aligned}$$

where  $\Delta \mathbf{v}_l(i)$  is the incremental velocity vector over the  $i$ th interval within the  $l$  cycle [The sum of  $\Delta \mathbf{v}_l(1)$ ,  $\Delta \mathbf{v}_l(2)$ , and  $\Delta \mathbf{v}_l(3)$  equals  $\mathbf{v}(t)$  in Eq. (27) over an  $l$  cycle (integral limits from  $l-1$  to  $l$ ); and  $\Delta \mathbf{v}_l$ ,  $\mathbf{v}_{l-1}$  is identical to the Eq. (33) definitions. Using Eq. (10), we can write for the combined velocity rotation compensation/sculling algorithm:

$$\Delta \hat{\mathbf{v}}_{\text{rot/scul}}(m) = \Delta \hat{\mathbf{v}}_{\text{rot}}(m) + \Delta \hat{\mathbf{v}}_{\text{scul}}(m) \quad (37)$$

where  $\Delta \hat{\mathbf{v}}_{\text{rot}}(m)$  is equivalent to Eq. (7):

$$\Delta \hat{\mathbf{v}}_{\text{rot}}(m) = \frac{1}{2} \boldsymbol{\alpha}(m) \times \mathbf{v}(m) \quad (38)$$

Substituting Eqs. (38) and (36) in Eq. (37) obtains

$$\begin{aligned}
 \Delta \hat{\mathbf{v}}_{\text{rot/scul}}(m) &= \frac{1}{2} \boldsymbol{\alpha}(m) \times \mathbf{v}(m) \\
 &\quad + \frac{1}{2} \sum_{l=2}^L [(\boldsymbol{\alpha}_{l-1} \times \Delta \mathbf{v}_l) + (\mathbf{v}_{l-1} \times \Delta \boldsymbol{\alpha}_l)] \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \boldsymbol{\alpha}_l(1) + \frac{27}{20} \Delta \boldsymbol{\alpha}_l(2) \right] \times \Delta \mathbf{v}_l(3) \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \mathbf{v}_l(1) + \frac{27}{20} \Delta \mathbf{v}_l(2) \right] \times \Delta \boldsymbol{\alpha}_l(3) \quad (39)
 \end{aligned}$$

The rotation compensation term in Eq. (39) is also

$$\frac{1}{2} \boldsymbol{\alpha}(m) \times \mathbf{v}(m) \equiv \frac{1}{2} \boldsymbol{\alpha}_L \times \mathbf{v}_L = \frac{1}{2} \sum_{l=1}^L [(\boldsymbol{\alpha}_l \times \mathbf{v}_l) - (\boldsymbol{\alpha}_{l-1} \times \mathbf{v}_{l-1})] \quad (40)$$

where it is recognized that all terms in the series expansion cancel except for the first and last and that  $\boldsymbol{\alpha}_{l-1}$  and  $\mathbf{v}_{l-1}$  are zero for  $l=1$ . From the preceding definitions for  $\boldsymbol{\alpha}_l$ ,  $\mathbf{v}_l$ ,  $\boldsymbol{\alpha}_{l-1}$ ,  $\mathbf{v}_{l-1}$ ,  $\Delta \boldsymbol{\alpha}_l$ , and  $\Delta \mathbf{v}_l$

$$\boldsymbol{\alpha}_l = \boldsymbol{\alpha}_{l-1} + \Delta \boldsymbol{\alpha}_l, \quad \mathbf{v}_l = \mathbf{v}_{l-1} + \Delta \mathbf{v}_l \quad (41)$$

Substituting Eq. (41) in Eq. (40) obtains

$$\begin{aligned}
 \frac{1}{2} \boldsymbol{\alpha}(m) \times \mathbf{v}(m) &= \frac{1}{2} \sum_{l=1}^L [(\boldsymbol{\alpha}_l \times \mathbf{v}_l) - (\boldsymbol{\alpha}_{l-1} \times \mathbf{v}_{l-1})] \\
 &= \frac{1}{2} \sum_{l=1}^L \{[(\boldsymbol{\alpha}_{l-1} + \Delta \boldsymbol{\alpha}_l) \times (\mathbf{v}_{l-1} + \Delta \mathbf{v}_l)] - (\boldsymbol{\alpha}_{l-1} \times \mathbf{v}_{l-1})\} \\
 &= \frac{1}{2} \sum_{l=1}^L [(\boldsymbol{\alpha}_{l-1} \times \Delta \mathbf{v}_l) + (\Delta \boldsymbol{\alpha}_l \times \mathbf{v}_{l-1}) + (\Delta \boldsymbol{\alpha}_l \times \Delta \mathbf{v}_l)] \\
 &= \frac{1}{2} \sum_{l=2}^L [(\boldsymbol{\alpha}_{l-1} \times \Delta \mathbf{v}_l) - (\mathbf{v}_{l-1} \times \Delta \boldsymbol{\alpha}_l)] \\
 &\quad + \frac{1}{2} \sum_{l=1}^L (\Delta \boldsymbol{\alpha}_l \times \Delta \mathbf{v}_l) \quad (42)
 \end{aligned}$$

Finally, substitution of Eq. (42) in Eq. (39) yields an alternate to Eq. (39) for the combined velocity rotation compensation/sculling algorithm:

$$\begin{aligned}
 \Delta \hat{\mathbf{v}}_{\text{rot/scul}}(m) &= \sum_{l=2}^L (\boldsymbol{\alpha}_{l-1} \times \Delta \mathbf{v}_l) + \frac{1}{2} \sum_{l=1}^L (\Delta \boldsymbol{\alpha}_l \times \Delta \mathbf{v}_l) \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \boldsymbol{\alpha}_l(1) + \frac{27}{20} \Delta \boldsymbol{\alpha}_l(2) \right] \times \Delta \mathbf{v}_l(3) \\
 &\quad + \sum_{l=1}^L \left[ \frac{9}{20} \Delta \mathbf{v}_l(1) + \frac{27}{20} \Delta \mathbf{v}_l(2) \right] \times \Delta \boldsymbol{\alpha}_l(3) \quad (43)
 \end{aligned}$$

The combined velocity rotation compensation/sculling algorithm counterpart to the Ref. 2 (algorithm 3) coning algorithm [Eq. (34)] is derived in Ref. 6 [Eqs. (7) and (21) with the Table 1, algorithm 2 coefficients] and is identical to Eq. (43).

## Conclusion

The generic equivalency that exists between coning and sculling integrals has been exploited to produce a simple mathematical formula that converts already derived coning algorithms to their sculling algorithm counterparts. Because a substantial number of coning algorithms already exist, conversion to their sculling algorithm equivalents using the process presented here is far simpler than independently deriving new sculling algorithms from scratch.

## Acknowledgment

The author wishes to express her appreciation to Paul G. Savage of Strapdown Associates, Inc., for proposing the generic cross product vector integral equivalencies used in this paper.

## References

- 1 Miller, R. B., "A New Strapdown Attitude Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 4, 1993, pp. 287–291.

- <sup>2</sup>Ignagni, M. B., "Efficient Class of Optimized Coning Compensation Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 424-429.
- <sup>3</sup>Ignagni, M. B., "Optimal Strapdown Attitude Integration Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 363-369.
- <sup>4</sup>Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 19-28.
- <sup>5</sup>Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms," *Journal of*

*Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 208-221.

<sup>6</sup>Ignagni, M. B., "Duality of Optimal Strapdown Sculling and Coning Compensation Algorithms," *Journal of the Institute of Navigation*, Vol. 45, No. 2, 1998, pp. 85-95.

<sup>7</sup>Mark, J. G., and Tazartes, D. A., "On Sculling Algorithms," *3rd International Conference on Integrated Navigation Systems, Pt. 2*, Litton Guidance and Control Systems, Woodland Hills, CA, May 1996.

<sup>8</sup>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.