AIAA 81-4311

Strapdown Attitude Algorithms from a Geometric Viewpoint

Richard McKern* and Howard Musoff†
The Charles Stark Draper Laboratory, Inc., Cambridge, Mass.

This paper derives various well-known strapdown inertial system attitude algorithms using a geometric viewpoint based on the Goodman-Robinson theorem. This theorem describes three-dimensional rotation kinematics of a rigid body. The attitude algorithms derived are the third-order quaternion and direction cosine matrix, the third-order quaternion modified to compute the cross-product term at twice the update rate and the computationally partitioned algorithm using the rotation vector and an associated quaternion. The distinguishing features of these algorithms are readily apparent from the unified derivations. In particular, the assumptions made in the computation of areas on a unit sphere using the Goodman-Robinson theorem illuminates the performance limitations of these algorithms and should be of use in deriving more efficient algorithms.

Introduction

THE purpose of this paper is to review various commonly used attitude algorithm development approaches and to more easily and intuitively explain their algebraic formulation by using a unifying geometric viewpoint.

The attitude algorithms dealt with here are mathematical procedures with which incremental attitude data is continually processed to compute a transformation from one known coordinate frame to another. In a strapdown system, this algorithm accepts incremental rotation angles measured directly by a gyroscope. This gyroscope data describes orientation changes of the gyroscope mounting frame with respect to an inertial frame. The data from three orthogonally mounted devices can be used to describe the attitude between these two frames. The transformation representing the output of the attitude algorithm is available for transforming body accelerometer data into an inertially referenced navigation frame and can be used for digital flight control processing as well as other body attitude or attitude rate requirements.

The two most commonly used transformations are direction cosine matrices and quaternions of rotation. These expressions satisfy linear differential equations where the input form is angular rate and not incremental angles as provided by the gyroscope and its associated electronics. There are a number of numerical methods available for solving these differential equations using incremental angle inputs. Methods which can be applied to this problem include Taylor series expansions, Runge-Kutta algorithms, and predictor-corrector algorithms. Of these methods, the Taylor series expansion lends itself directly to use of incremental angles.[1]

Any digital attitude algorithm can be thought of as a computation of an attitude transformation from a finite sequence of incremental slewing motions at a fixed update rate. The geometric approach views this computation as an interpolation of the true angular motion over the sequence of slewing increments, which can be regarded as an approximation of areas on a unit sphere. These areas are related to the incremental rotation vector and to the angular increments from the gyros by use of the Goodman-Robinson theorem.[2,3]

Goodman-Robinson Theorem

In a rigid body (with a defined orthogonal triad of body axes) undergoing an arbitrary angular rotation, the component ϕ_i (along body axis i) of the rotation vector ϕ is related to the integral of the component ω_i of the angular velocity vector ω through the equation

$$\phi_i = \int_{t_0}^{t_f} \omega_i \mathrm{d}t + A_i \qquad (1)$$

1) A_i is the area on the surface of a unit sphere enclosed by the curve traced out on the surface by axis i and an arc on the sphere given by a pure slew that connects the endpoints of this curve. This connecting arc is traced by the body axes undergoing a constant slew about the rotation axis needed to return the body axes to their initial orientations as illustrated in Fig. 1. The sphere shown is centered at the origin at the body axis triad.

2) $$\int_{t_0}^{t_f} \omega_i \mathrm{d}t < 2\pi$$

3) The curve traced out by axis i does not intersect itself. Pure slewing motion is described as an angular motion about a fixed axis of rotation in the body such that A_i will be zero. This theorem is expressed for slewing motion as

$$\phi_i = \int_{t_0}^{t_f} \omega_i \mathrm{d}t \qquad (A_i = 0 \quad \text{for} \quad \text{pure slewing}) \qquad (2)$$

Geometric Interpretation of Digital Attitude Algorithms

A geometric viewpoint is introduced as an aid in a practical understanding of digital attitude algorithms and how attitude errors can develop. Digital attitude algorithm execution in a strapdown inertial system using integrating gyroscopes involves the processing of incremental angles. Each finite incremental angle update can be physically interpreted as an incremental slew during which each body axis traces out an incremental slewing arc. The true incremental rotation vector component is given by the sum of the incremental slewing angles, and the area bounded both by the slewing arc and the curve on the unit sphere traced out by the body axis under consideration. A fundamental function of the attitude algorithm is to make correct computations of the incremental rotation vectors based on the elementary slews and appropriate unit sphere areas. The areas are computed by approximating the true area, that is, computing an area based on elementary slewing data. Other functions of the algorithm are to convert the incremental rotation vector into its quaternion or direction cosine equivalent and update the total angle quaternion or direction cosine matrix with the most recent incremental data.

Figure 2 illustrates the geometric process of computing the incremental rotation vector using incremental attitude and

Received Aug. 8, 1980; revision received Feb. 27, 1981.

*Section Chief. Member AIAA.

†Staff Engineer. Member AIAA.

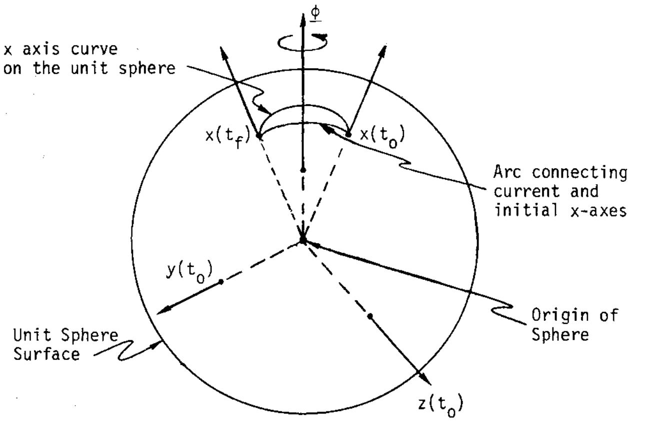

Fig. 1 Illustration of Goodman-Robinson theorem applied to *x*-body axis.

estimated areas on a unit sphere. This illustration is for the *x*-body axis but is equally appropriate to each of the three computational axes. This geometric interpretation of the attitude algorithm based on the properties of the Goodman-Robinson theorem will now be reinforced by deriving common algorithms using these geometric principles.

Derivation of Two Conventional Attitude Algorithms

First, we will derive a quaternion third-order algorithm using the geometric viewpoint as well as Taylor series expansions. The incremental rotation vector $\boldsymbol{\Delta\phi}$ during a single update period can be expressed as

$$\boldsymbol{\Delta\phi} = \boldsymbol{\Delta\theta} + \boldsymbol{A} \tag{3}$$

where each component of $\boldsymbol{A}$ is the appropriate area presented in the Goodman-Robinson theorem and $\boldsymbol{\Delta\theta}$ is the incremental angle data obtained from the gyroscopes.

An expression for the incremental quaternion can be given from its definition by

$$\Delta q = \left(\cos\frac{\Delta\phi}{2}, \frac{\boldsymbol{\Delta\phi}}{\Delta\phi}\sin\frac{\Delta\phi}{2}\right) \triangleq (\Delta\lambda, \boldsymbol{\Delta\rho}) \quad \Delta\phi = (\boldsymbol{\Delta\phi}\cdot\boldsymbol{\Delta\phi})^{1/2} \tag{4}$$

A Taylor series expansion of Eq. (4) to third order is given by

$$\Delta\lambda = \cos\frac{\Delta\phi}{2} \cong 1 - \frac{\Delta\phi^2}{8} = 1 - \frac{\boldsymbol{\Delta\phi}\cdot\boldsymbol{\Delta\phi}}{8}$$

$$\boldsymbol{\Delta\rho} = \frac{\boldsymbol{\Delta\phi}}{\Delta\phi}\sin\frac{\Delta\phi}{2} \cong \frac{\boldsymbol{\Delta\phi}}{\Delta\phi}\left(\frac{\Delta\phi}{2} - \frac{\Delta\phi^3}{48}\right) = \frac{\boldsymbol{\Delta\phi}}{2} - \frac{\boldsymbol{\Delta\phi}(\boldsymbol{\Delta\phi}\cdot\boldsymbol{\Delta\phi})}{48} \tag{5}$$

Now, assuming the area vector ($\boldsymbol{A}$) is dependent on at least the square of the $\boldsymbol{\Delta\theta}$ magnitude, and using Eqs. (3) and (5) to derive the quaternion expression, we obtain to third order

$$\Delta\lambda \cong 1 - \frac{\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta}}{8} \qquad \boldsymbol{\Delta\rho} \cong \frac{\boldsymbol{\Delta\theta}}{2} - \frac{\boldsymbol{\Delta\theta}(\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta})}{48} + \frac{\boldsymbol{A}}{2} \tag{6}$$

The steps leading to Eq. (6) detail the method of converting the incremental rotation vector to an incremental quaternion. The area vector ($\boldsymbol{A}$) remains to be determined. Each component (A_i) is an area on the unit sphere. Assuming small increments of $\boldsymbol{\Delta\theta}$, this spherical area can be assumed planar. Figure 3 illustrates the planar geometry for computation of the A_x component. In this figure, $\boldsymbol{\Delta\theta}_n^{yz}$ is a two-dimensional vector at the present step n given by

$$\boldsymbol{\Delta\theta}_n^{yz} = \begin{bmatrix} \Delta\theta_y(n) \\ \\ \Delta\theta_z(n) \end{bmatrix} \tag{7}$$

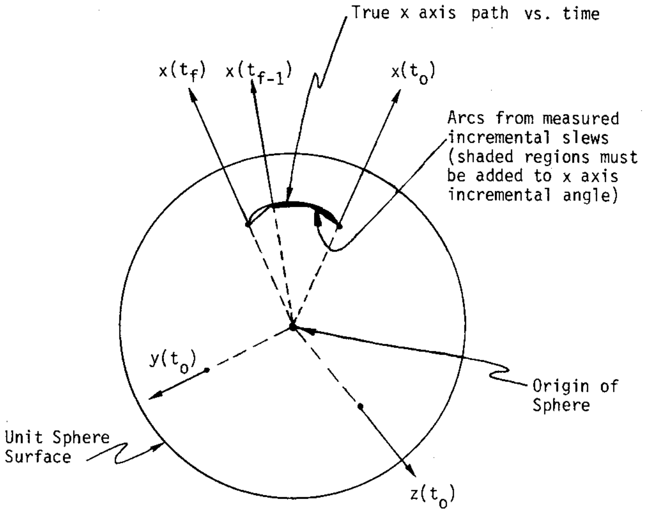

Fig. 2 Interpolation of a body axis trace on a unit sphere applied to *x*-body axis.

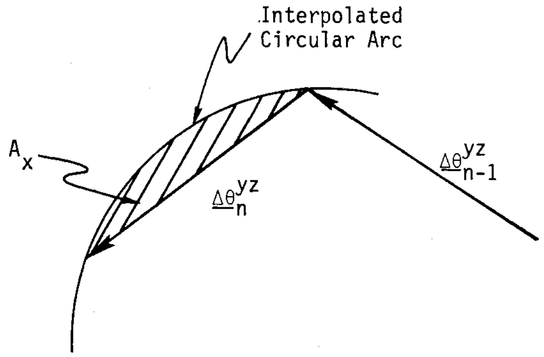

Fig. 3 Computation of area on a plane approximating the portion of the sphere illustrated for the *x*-body axis.

In a similar manner, $\boldsymbol{\Delta\theta}_{(n-1)}^{yz}$ is a two-dimensional vector at the previous step. A_x can now be calculated from these two data sets for an appropriate interpolated curve approximating the correct path of the tip of the *x*-body axis on the unit sphere. It is proven in a straightforward manner in the Appendix that a circular arc assumption for the interpolated curve results in an area estimate of

$$A_x \cong [\boldsymbol{\Delta\theta}_{n-1}^{yz} \times \boldsymbol{\Delta\theta}_n^{yz}]/12 \tag{8a}$$

Thus, for all three areas,

$$\boldsymbol{A} \cong (\boldsymbol{\Delta\theta}_{n-1} \times \boldsymbol{\Delta\theta}_n)/12 \tag{8b}$$

The resultant quaternion algorithm is now given by

$$\Delta\lambda \cong 1 - \frac{\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta}}{8}$$

$$\boldsymbol{\Delta\rho} \cong \frac{\boldsymbol{\Delta\theta}}{2} - \frac{\boldsymbol{\Delta\theta}(\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta})}{48} + \frac{1}{24}\boldsymbol{\Delta\theta}_{n-1} \times \boldsymbol{\Delta\theta}_n \tag{9}$$

which is exactly the same as the third-order algorithm derived by classical means.

The direction cosine matrix equation (equivalent to the quaternion equation described above) can be developed using a similar line of reasoning. C is defined in terms of the rotation vector as[1]:

$$C = I + N\sin\phi + N^2(1-\cos\phi) \tag{10}$$

where I is the unit matrix and N is the skew symmetric matrix as follows:

$$N \triangleq \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} = \frac{1}{\phi} \begin{bmatrix} 0 & -\phi_z & \phi_y \\ \phi_z & 0 & -\phi_x \\ -\phi_y & \phi_x & 0 \end{bmatrix}$$

if

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \text{ and } \boldsymbol{\phi} = \boldsymbol{n}\phi \neq \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}$$

Thus, as illustrated for the quaternion, ΔC can be found from the definition in Eq. (1) by

$$\Delta C = I + \frac{1}{\Delta\phi}\Gamma(\boldsymbol{\Delta\phi})\sin\Delta\phi + \frac{1}{\Delta\phi^2}\Gamma^2(\boldsymbol{\Delta\phi})(1-\cos\Delta\phi) \tag{11}$$

when

$$\Gamma(\boldsymbol{\Delta\phi}) \triangleq \begin{bmatrix} 0 & -\Delta\phi_z & \Delta\phi_y \\ \Delta\phi_z & 0 & -\Delta\phi_x \\ -\Delta\phi_y & \Delta\phi_x & 0 \end{bmatrix} \qquad \Delta\phi \triangleq (\boldsymbol{\Delta\phi}\cdot\boldsymbol{\Delta\phi})^{1/2}$$

Again, using Eq. (3) and the definition of ΔC in Eq. (11) along with the appropriate truncated Taylor expansions and circular arc assumptions, it can be shown that

$$\Delta C \cong I + \left(1 - \frac{\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta}}{6}\right)\Gamma(\boldsymbol{\Delta\theta}) + \frac{1}{2}\Gamma^2(\boldsymbol{\Delta\theta}) + \frac{1}{12}\Gamma(\boldsymbol{\Delta\theta}_{n-1}\times\boldsymbol{\Delta\theta}_n) \tag{12}$$

which is again identical to the classical derivation using a Taylor expansion to the third order of the direction cosine matrix attitude algorithm.

Partitioned Algorithm Mechanization (Ref. 4)

In order to improve algorithm efficiency with negligible performance degradation, certain update rate sensitive equation repartitioning is suggested along the lines of a technique initially proposed by Bortz.[5] The algorithm computation is broken into separate sections. A high update rate section closely approximates the true motion by rapidly processing elementary slews of $\boldsymbol{\Delta\theta}_n$ into a small-angle rotation vector ($\boldsymbol{\phi}_n$). Then, at a significantly lower rate, the rotation vector is incorporated into a whole-angle representation quaternion and is reinitialized to zero. The rate is selected, by knowledge of the application, to ensure that $\boldsymbol{\phi}_n$ remains a small angle. The major advantage of this two-section mechanization is that it allows a high-rate loop to maintain the incremental motion closely associated with true motion, while allowing the computationally more burdensome quaternion of rotation update to be processed at much lower rate. Consider the motion to be composed only of elementary slews. Approximation of the true curve area is handled exactly as previously described. In the partitioned algorithm development, we are concerned with updating a rotation vector ($\boldsymbol{\phi}_{n-1}$) after a slew of $\boldsymbol{\Delta\theta}_n$. An exact expression for the desired updated rotation vector $\boldsymbol{\phi}_n$ is easily obtained using the rules of quaternion multiplication and is given by

$$\boldsymbol{\phi}_n = \frac{\phi_n}{\sin(\phi_n/2)}\left[\cos\frac{\Delta\theta_n}{2}\,\frac{\boldsymbol{\phi}_{n-1}}{\phi_{n-1}}\sin\frac{\phi_{n-1}}{2} + \cos\frac{\phi_{n-1}}{2}\,\frac{\boldsymbol{\Delta\theta}_n}{\Delta\theta_n}\sin\frac{\Delta\theta_n}{2} + \frac{\sin(\phi_{n-1}/2)\sin(\Delta\theta_n/2)}{\phi_{n-1}\Delta\theta_n}(\boldsymbol{\phi}_{n-1}\times\boldsymbol{\Delta\theta}_n)\right] \tag{13}$$

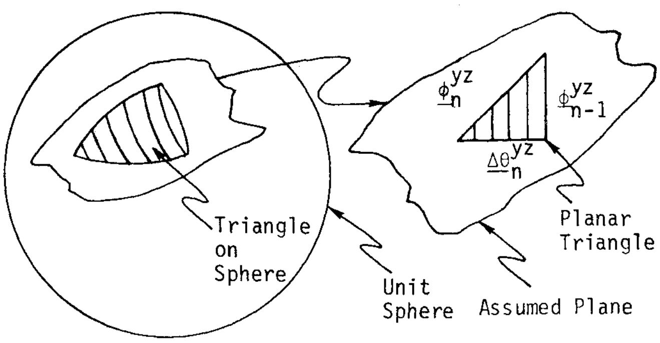

Fig. 4 Approximation by a planar triangle.

with

$$\Delta\theta \triangleq (\boldsymbol{\Delta\theta}\cdot\boldsymbol{\Delta\theta})^{1/2}$$

Here we are assuming that both ϕ_n and $\Delta\theta_n$ are small-angle magnitudes such that

$$\sin(\phi_n/2) \cong \phi_n/2 \qquad \cos(\phi_n/2) \cong 1$$

$$\sin(\Delta\theta_n/2) \cong \Delta\theta_n/2 \quad \cos(\Delta\theta_n/2) \cong 1 \tag{14}$$

Using Eqs. (13) and (14), the rotation vector may be simplified to

$$\boldsymbol{\phi}_n = \boldsymbol{\phi}_{n-1} + \boldsymbol{\Delta\theta}_n + \tfrac{1}{2}\boldsymbol{\phi}_{n-1}\times\boldsymbol{\Delta\theta}_n \tag{15}$$

If it is assumed that only elementary slews of $\boldsymbol{\Delta\theta}_n$ occur each update, then Eq. (15) is readily interpreted by means of the Goodman-Robinson theorem. Here the area vector $\boldsymbol{A}$ is approximated by

$$\boldsymbol{A} \cong \tfrac{1}{2}\boldsymbol{\phi}_{n-1}\times\boldsymbol{\Delta\theta}_n \tag{16a}$$

Clearly, each component of the above $\boldsymbol{A}$ is an area of a triangle defined on a plane. We have approximated the areas of triangles on the sphere by assuming that the spherical triangular areas which are traced out on the unit surface over the intervals during which $\boldsymbol{\phi}_n$ is calculated can be approximated by planar areas (see Fig. 4).

Here the pertinent area for the x-body axis is given by

$$A_x = \tfrac{1}{2}[\boldsymbol{\phi}^{yz}_{n-1}\times\boldsymbol{\Delta\theta}^{yz}_n] \tag{16b}$$

where

$$\boldsymbol{\phi}^{yz}_{n-1} \triangleq \begin{bmatrix} \phi_y(n-1) \\ \phi_z(n-1) \end{bmatrix} \qquad \boldsymbol{\Delta\theta}^{yz}_n = \begin{bmatrix} \Delta\theta_y(n) \\ \Delta\theta_z(n) \end{bmatrix}$$

To complete the derivation of the partitioned algorithm, the area vector given in Eqs. (16) must be modified by the addition of the interpolated area given in Eq. (8b). The complete equation for the incremental rotation vector in the partitioned algorithm is thus given by

$$\boldsymbol{\phi}_n = \boldsymbol{\phi}_{n-1} + \boldsymbol{\Delta\theta}_n + \tfrac{1}{2}\boldsymbol{\phi}_{n-1}\times\boldsymbol{\Delta\theta}_n + (1/12)\boldsymbol{\Delta\theta}_{n-1}\times\boldsymbol{\Delta\theta}_n \tag{17}$$

The algorithm which incorporates a high-rate update of the incremental rotation vector at time t is then

$$\boldsymbol{\phi}_n = \boldsymbol{\phi}_{n-1} + \boldsymbol{\Delta\theta}_n + [\tfrac{1}{2}\boldsymbol{\phi}_{n-1} + (1/12)\boldsymbol{\Delta\theta}_{n-1}]\times\boldsymbol{\Delta\theta}_n \tag{18}$$

The low-rate computation then consists solely of forming $\boldsymbol{\phi}_n$ into a quaternion representation $q(\Delta t)$ and updating the whole-angle rotation quaternion to the current time t while resetting $\boldsymbol{\phi}_n$ to zero for reinitiation of the high-rate computation. Thus, for the low rate,

$$q(t) = q(t-\Delta t)\,q(\Delta t) \tag{19}$$

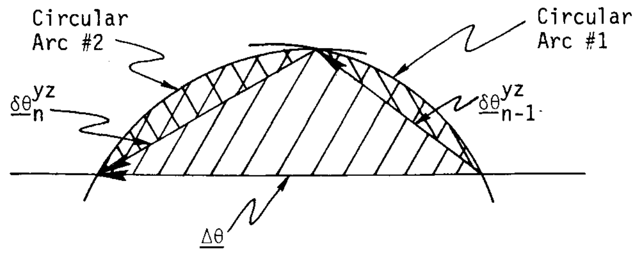

Fig. 5 Geometry applicable to twice frequency cross-product algorithm.

Coning environment simulation has shown the partitioned algorithm using a 100 Hz high-rate iteration and 20 Hz low-rate iteration to be similar in performance to a full third-order quaternion (performed at 100 Hz) with a 1-deg coning angle out to at least a 5-Hz coning frequency. It has also been found that degradation due to word length and quantum angle sizing limitations is similar in the two mechanizations.

In practice, the fast loop attitude is also used for small-angle corrections to maintain body frame acceleration accuracy compatible with the slower velocity transformation rate of the partitioned algorithm implementation. The partitioned algorithm can maintain accuracies compatible with the conventional third-order quaternion at about half the real-time computational burden. Memory requirements are similar to the conventional algorithm.

From the geometric development, we see that the considerable savings in computation possible with the partitioned algorithm are a direct result of the approximation depicted in Fig. 4; that is, the area of a planar triangle is used to approximate the area of a spherical triangle.

Twice-Frequency Cross-Product Algorithm Variation

A slight variation in the third-order quaternion, which formulates the cross-product area at twice the update rate, has been presented by Wilcox.[6]

We can also use the geometric approach to describe this development which uses the angular increments taken at twice the algorithm update rate to calculate the cross-product term only. The remaining terms are formed from the sum of the two angular increments. Here, the incremental quaternion is given by

$$\Delta\lambda \cong 1 - \frac{\Delta\theta\cdot\Delta\theta}{8} \qquad \Delta\rho \cong \frac{\Delta\theta}{2} - \frac{\Delta\theta(\Delta\theta\cdot\Delta\theta)}{48} + \frac{1}{3}\delta\theta_{n-1}\times\delta\theta_n \tag{20}$$

where the cross product is formed in succeeding updates from $\delta\theta_n$ and $\delta\theta_{n-1}$, $\delta\theta_{n+2}$ and $\delta\theta_{n+1}$, etc., and

$$\Delta\theta = \delta\theta_n + \delta\theta_{n-1} \tag{21}$$

During the next update,

$$\Delta\theta = \delta\theta_{n+2} + \delta\theta_{n+1} \tag{22}$$

Derivation of the $\Delta\theta$ terms in Eq. (20) is obviously exactly the same as before. However, it remains to justify the ⅓-coefficient in the cross product terms $⅓\delta\theta_{n-1}\times\delta\theta_n$. Figure 5 illustrates the applicable geometry.

The interpolated area consists of the region bounded by the circular arcs and $\Delta\theta$. The area of the triangular portion of this region (bounded by $\delta\theta_n^{yz}$, $\delta\theta_{n-1}^{yz}$, and $\Delta\theta$) is given by

$$A_x' = ½[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}] \tag{23}$$

We have seen that the area of the region bounded by the circular arc 2 and $\delta\theta_n^{yz}$ can be approximated by $[\delta\theta^{yz}\times\delta\theta_n^{yz}]/12$.

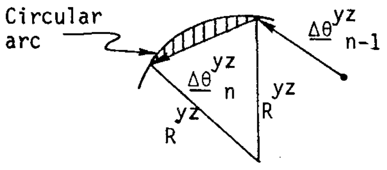

Fig. A-1 Interpolation with a circular arc.

It can be shown that the area of the region bounded by circular arc 1 and $\delta\theta_{n-1}^{yz}$ can also be approximated by $[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}]/12$. Then the two crosshatched areas in Fig. 5 are approximated by $[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}]/6$ which, when added to A_x' in Eq. (23), yields the total interpolated area

$$A_x = ½[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}] + 1/6[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}] = ⅔[\delta\theta_{n-1}^{yz}\times\delta\theta_n^{yz}] \tag{24}$$

The associated area vector A is therefore given by

$$A = ⅔(\delta\theta_{n-1}\times\delta\theta_n) \tag{25}$$

Substitution of Eq. (25) into Eq. (6) results in the required term, $A/2$, given by

$$A/2 = ⅓(\delta\theta_{n-1}\times\delta\theta_n) \tag{26}$$

in Eq. (20).

In this mechanization, memory requirements are about the same as the conventional quaternion. This modified quaternion will improve performance in a coning environment with the same update rate as in the conventional third-order algorithm.

Conclusions

Using a unified geometric approach, we have derived several commonly used strapdown attitude algorithms. The algorithms illustrated are the third-order quaternion and direction cosine matrix, the third-order quaternion modified to compute the cross-product term at twice the update rate, and the computationally partitioned algorithm using the rotation vector and an associated quaternion. Estimates of algorithm performance and computer burden, relative to the conventional third-order algorithms, are given for the partitioned and twice-frequency algorithms.

The geometric approach shows how these properties of different algorithms arise from the computational assumptions made during the algorithm formulation.

The modified third-order quaternion algorithm for computing the cross-product term at twice update rate will improve performance in a coning environment with update rate as in the conventional third-order algorithm.

Coning environment simulation has shown the partitioned algorithm comparable in performance to a full third-order algorithm with a significantly lower computational burden.

The geometric approach developed here should provide insight in developing new algorithms and in making comparative judgments in a specific application.

Appendix: Interpolation with a Circular Arc Computation of Area on a Plane

Figure A-1 illustrates the geometry for fitting a circular arc to the path of slews. R^{yz} is the radius of curvature computed from $\Delta\theta_n^{yz}$ and $\Delta\theta_{n-1}^{yz}$.

The curvature, K, of a plane curve is given by the following definition with t as a parameter of the curve:

$$K = \frac{\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)\left(\frac{\mathrm{d}^2v}{\mathrm{d}t^2}\right) - \left(\frac{\mathrm{d}v}{\mathrm{d}t}\right)\left(\frac{\mathrm{d}^2u}{\mathrm{d}t^2}\right)}{\left[\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}v}{\mathrm{d}t}\right)^2\right]^{3/2}} \tag{A1}$$

where the curve is defined by

$$v(t)=f(u(t)) \tag{A2}$$

A numerical approximation to the curvature based only on the latest three points, (u,v), can be obtained from

$$K\simeq \frac{\left(\frac{\Delta u_n}{\Delta t}\right)\left(\frac{\Delta v_n-\Delta v_{n-1}}{\Delta t^2}\right)-\left(\frac{\Delta v_n}{\Delta t}\right)\left(\frac{\Delta u_n-\Delta u_{n-1}}{\Delta t^2}\right)}{(\Delta u_n^2+\Delta v_n^2)^{3/2}(\Delta t)^{-3}}$$

$$=\frac{\Delta v_n\Delta u_{n-1}-\Delta u_n\Delta v_{n-1}}{(\Delta u_n^2+\Delta v_n^2)^{3/2}} \tag{A3}$$

where

$$\Delta v_n=v(t_n)-v(t_{n-1}) \qquad \Delta v_{n-1}=v(t_{n-1})-v(t_{n-2})$$

and similarly for Δu_n, Δu_{n-1}.

If we identify Δu_n with $\Delta\theta_{yn}$ and Δv_n with $\Delta\theta_{zn}$, then the curvature is approximated by

$$K^{yz}\simeq |\Delta\boldsymbol{\theta}_n^{yz}\times\Delta\boldsymbol{\theta}_{n-1}^{yz}|/(\Delta\theta_n^{yz})^3 \tag{A4}$$

where $\Delta\boldsymbol{\theta}_n^{yz}$ and $\Delta\boldsymbol{\theta}_{n-1}^{yz}$ are defined by Eq. (7),

$$\Delta\theta_n^{yz}\triangleq(\Delta\boldsymbol{\theta}_n^{yz}\cdot\Delta\boldsymbol{\theta}_n^{yz})^{1/2} \tag{A5}$$

and K^{yz} denotes the curvature in the yz plane.

The radius of curvature R^{yz} in Fig. A-1 is therefore given by

$$R^{yz}=\frac{1}{K^{yz}}\simeq\frac{(\Delta\theta_n^{yz})^3}{|\Delta\boldsymbol{\theta}_n^{yz}\times\Delta\boldsymbol{\theta}_{n-1}^{yz}|} \tag{A6}$$

Elementary geometric considerations lead to the following expression for the area A_x of the shaded segment in Fig. A-1.

$$A_x=(R^{yz})^2\arcsin(\Delta\theta_n^{yz}/2R^{yz})-\tfrac{1}{2}\Delta\theta_n^{yz}R^{yz}$$

$$\times\sqrt{1-(\Delta\theta_n{}^{yz}/2R^{yz})^2} \tag{A7}$$

where the first term on the right-hand side is the expression for the area of the appropriate sector of the circle of radius R^{yz}, and the second term is the expression for the area of the appropriate wedge.

If $R^{yz}\gg\Delta\theta_n^{yz}$ (i.e., $\Delta\boldsymbol{\theta}_n^{yz}$ and $\Delta\boldsymbol{\theta}_{n-1}^{yz}$ are roughly in the same direction), then

$$A_x\simeq(R^{yz})^2\left(\frac{\Delta\theta_n^{yz}}{2R^{yz}}+\frac{1}{6}\left(\frac{\Delta\theta_n^{yz}}{2R^{yz}}\right)^3\right)-\frac{1}{2}\Delta\theta_n^{yz}R^{yz}\left(1-\frac{1}{2}\left(\frac{\Delta\theta_n^{yz}}{2R^{yz}}\right)^2\right)$$

$$=\frac{1}{48}\frac{(\Delta\theta_n^{yz})^3}{R^{yz}}+\frac{1}{16}\frac{(\Delta\theta_n^{yz})^3}{R^{yz}}=\frac{1}{12}\frac{(\Delta\theta_n^{yz})^3}{R^{yz}} \tag{A8}$$

Substituting Eq. (A6) into Eq. (A8) yields finally,

$$A_x\simeq|\Delta\boldsymbol{\theta}_n^{yz}\times\Delta\boldsymbol{\theta}_{n-1}^{yz}|/12 \tag{A9}$$

which can be interpreted as 1/12 the area of the parallelogram defined by $\Delta\boldsymbol{\theta}_n^{yz}$ and $\Delta\boldsymbol{\theta}_{n-1}^{yz}$.

Acknowledgment

The authors thank Martin Landey for editing and Jean Avery for typing this text.

References

[1] McKern, R.A., "A Study of Transformation Algorithms for Use in a Digital Computer," MIT Instrumentation Laboratory Rept. T-493, 1968.

[2] Goodman, L.E. and Robinson, A.R., "Effect of Finite Rotations on Gyroscope Sensing Devices," *Journal of Applied Mechanics,* Vol. 25, June 1968, pp. 210-213.

[3] Broxmeyer, C., *Inertial Navigation Systems,* McGraw-Hill, New York, 1964, p. 31.

[4] Musoff, H., "Partitioned Strapdown Algorithm: Part II," ISS Memo No. 75-325, C.S. Draper Laboratory, Dec. 1975.

[5] Bortz, J., "A New Concept in Strapdown Inertial Navigation," Doctoral Thesis, MIT Dept. of Aeronautics and Astronautics, 1969.

[6] Wilcox, J.C., "A New Algorithm for Strapped Down Inertial Navigation," *IEEE Transactions on Aerospace and Electronic Systems,* AES-3, No. 5, Sept. 1967, pp. 796-802.