Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes cannot exceed 6 manuscript pages and 3 figures; a page of text may be substituted for a figure and vice versa. After informal review by the editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Singularity-free Extraction of a Quaternion from a Direction-Cosine Matrix

Allan R. Klumpp*
The Charles Stark Draper Laboratory, Cambridge, Mass.

USED as a four-parameter representation of a rotation, a quaternion affords considerable computational advantages compared to a direction-cosine matrix, its nine-parameter equivalent. To realize the quaternion's advantages, it is sometimes necessary to extract the quaternion from an existing matrix. As an example, for space shuttle steering the commanded attitude is readily computed as a direction-cosine matrix, whereas actual attitude is most compactly computed as a quaternion. Attitude error is computed as the quaternion product once the commanded-attitude quaternion is extracted from the corresponding matrix and conjugated. The virtues of an error *quaternion* are that the vector part coincides with the error axis and the length of the vector part is the sine of half the error angle. The half-angle property means length reaches maximum (unity) at 180° error, a very useful feature. The quaternion extraction algorithm is also used in several additional space shuttle flight-software applications.

Singularities in flight software are at best highly undesirable. Two algorithms for extracting a quaternion from a direction-cosine matrix have been found in the literature. The algorithm by Grubin[1] is the simplest, but it degrades in precision for rotation angles in the vicinity of 180° because it involves quotients in which both the dividend and divisor approach zero by subtraction of nearly equal numbers. At 180° it fails. The algorithm proposed by Rupp[2] and by Hendley[3] fails at 180° by yielding any one of eight solutions, six of which are incorrect. All algorithms must lose directional precision in the vicinity of null. Since the algorithm presented here is otherwise free from singularities, it has been adopted by the organizations involved in space shuttle guidance and control.

The vector part of the algorithm can be used separately as a compact singularity-free method for finding the eigenvector of a direction-cosine matrix.

A quaternion and a direction-cosine matrix are associated by their transformations,

$$\bar{V}^B = M_A^B \ \bar{V}^A \tag{1}$$

and

$$\begin{bmatrix} 0 \\ \bar{V}^B \end{bmatrix} = q_A^B \begin{bmatrix} 0 \\ \bar{V}^A \end{bmatrix} \left(q_A^B\right)^* \tag{2}$$

where $\bar{V}^A$ and $\bar{V}^B$ are either the same three-component vector expressed in frames A and B or are vectors of equal length within a single frame, M_A^B is the direction-cosine matrix, q_A^B is the quaternion, and $(q_A^B)^*$ is its conjugate. A matrix product is implied by Eq. (1) and quaternion products by Eq. (2).

Denoting the quaternion scalar element by q_0 and the vector elements by q_1, q_2, q_3, a direction-cosine matrix M is related to the quaternion q by

$$M = 2 \begin{bmatrix} q_1^2 + q_0^2 - ½ & q_1 q_2 - q_3 q_0 & q_1 q_3 + q_2 q_0 \\ q_2 q_1 + q_3 q_0 & q_2^2 + q_0^2 - ½ & q_2 q_3 - q_1 q_0 \\ q_3 q_1 - q_2 q_0 & q_3 q_2 + q_1 q_0 & q_3^2 + q_0^2 - ½ \end{bmatrix} \tag{3}$$

Two properties of use in extracting the quaternion from the matrix are: 1) a rotation quaternion is of unit length, and 2) a quaternion and its negative are interchangeable. A quaternion with positive scalar represents a rotation of 180° or less; its negative represents a rotation of opposite sense about the same axis through 360° minus the same angle. Since these rotations are equivalent, the quaternions are interchangeable. Thus, although there exist two quaternions corresponding to a single direction-cosine matrix, one quaternion can arbitrarily be selected with positive scalar corresponding to a rotation less than or equal to 180°.

Using the quaternion length property and choosing positive sign, the quaternion scalar element can be extracted from the matrix diagonal elements of Eq. (3) as

$$q_0 = +\sqrt{\frac{\text{tr } (M) + 1}{4}} \tag{4}$$

where tr (M) is the trace of the matrix.

An expression for the quaternion vector elements is immediately apparent in terms of the matrix off-diagonal elements. Let i, j, k index, in cyclic order, quaternion vector elements and matrix rows and columns. Then

$$q_k = (M_{j,i} - M_{i,j}) / 4 \ q_0 \tag{5}$$

This expression, presented by Grubin, reduces to 0/0 at 180° and is imprecise in the vicinity of 180° because numerator and denominator are both obtained by subtracting nearly equal quantities.

Imprecision in the vicinity of 180° can be overcome by computing the *magnitudes* of the quaternion vector elements in terms of the matrix diagonal elements

$$|q_i| = +\sqrt{M_{i,i}/2 + (1 - \text{tr } (M)) / 4} \tag{6}$$

Now the *signs* of the vector elements can be determined from the matrix off-diagonal elements. Fortunately, there are two expressions for doing this. The product of the quaternion scalar element and one quaternion vector element is

$$q_0 q_i = (M_{k,j} - M_{j,k}) / 4 \tag{7}$$

and the product of two quaternion vector elements is

$$q_i q_j = (M_{j,i} + M_{i,j}) / 4 \tag{8}$$

Since q_0 is positive, the sign of one vector element, q_i, is determined by Eq. (7) as

$$\text{sgn } (q_i) = \text{sgn } (M_{k,j} - M_{j,k}) \tag{9}$$

Received Feb. 17, 1976; revision received July 28, 1976.
Index categories: Navigation, Control, and Guidance Theory; Computer Technology and Computer Simulation Techniques.

*Staff Member. Presently at Jet Propulsion Laboratory, Pasadena, Calif.

With one vector element now known, the sign of another vector element is determined by Eq. (8) as

$$\text{sgn}\ (q_j) = \text{sgn}\ (q_i^2 q_j) = \text{sgn}\ [q_i(M_{j,i}+M_{i,j})] \qquad (10)$$

The form of Eq. (10) is chosen to avoid division because of the possibility that q_i is zero.

The signs produced by Eqs. (9) and (10) are indefinite in certain important cases. Indefiniteness cannot be avoided but it can be rendered inconsequential, as will be discussed next.

In the case of a 180° rotation, $q_0 = 0$, $M_{k,j} - M_{j,k} = 0$, and numerical imprecision causes Eq. (9) to produce indefinite signs. Thus, if the signs of all three vector elements are determined by cyclically permuting the indices of Eq. (9) as proposed by Rupp and Hendley, any one of eight solutions can occur, only two of which are correct. Two, not one, are correct because for a 180° rotation the vector part of the quaternion can arbitrarily be reversed. The remaining six (incorrect) solutions are reflections of the rotation axis about the planes of the coordinate frame. If Eq. (9) could be processed without numerical error, then for 180° rotations all three vectors elements would be assigned positive signs, also generally incorrect.

Because Eq. (9) is indefinite only when the rotation angle is 180°, in which case the direction of the rotation vector can arbitrarily be reversed, Eq. (9) can *always* be used to determine the sign of *one* vector element, say q_i. Then, given q_i, Eq. (10) can be used twice to determine the two remaining signs, provided q_i is not zero. If q_i is zero, then $M_{j,i} + M_{i,j} = 0$ and Eq. (10) is indefinite.

It is necessary and sufficient that q_i be the vector element of largest magnitude in order for this combination of Eqs. (9) and (10) to produce correct signs in all cases.

Necessity is established by the case in which two vector elements are zero and the rotation angle is other than 180° so that the sign of the nonzero element is not arbitrary. Then if q_i is one of the zero elements, Eq. (10) may incorrectly determine the sign of the nonzero element. Now because of numerical imprecision, two elements both nominally zero may differ. For example, one may be 1×10^{-17}, the other 2×10^{-17} while the nonzero element is ½. Equation (10) may fail if q_i is either "zero" element and both are avoided only if q_i is the vector element of largest magnitude.

Sufficiency is established because if q_i is the vector element of largest magnitude then the signs of the remaining two elements are correctly assigned regardless of their magnitudes. If all three vector elements are zero, indefinite signs are assigned without consequence.

The algorithm just developed, valid for all cases, consists of the following. First compute the scalar element by Eq. (4) and the magnitudes of the vector elements by Eq. (6). Then find i, j, k in cyclic order such that

$$|q_i| \geq |q_j|; \qquad |q_i| \geq |q_k| \qquad (11)$$

Then

$$q_i = \text{sgn}\ (M_{k,j} - M_{j,k})\ |q_i| \qquad (12)$$

$$q_j = \text{sgn}\ (q_i(M_{j,i} + M_{i,j}))\ |q_j| \qquad (13)$$

and

$$q_k = \text{sgn}\ (q_i(M_{k,i} + M_{i,k}))\ |q_k| \qquad (14)$$

where

$$\text{sgn}\ (A) \triangleq \begin{cases} -1, & A < 0 \\ +1, & A \geq 0 \end{cases} \qquad (15)$$

Equations (4) and (6) must be protected against square root failure because numerical imperfection can yield a negative argument which should be zero. Divide overflow can not occur because divisions by variables have been avoided.

References

[1] Grubin, C., "Derivation of the Quaternion Scheme via the Euler Axis and Angle," *Journal of Spacecraft and Rockets*, Vol. 7, Oct. 1970, pp. 1261-1263.

[2] Rupp, C. C., "Equations for Calculating Initial Values of the Four Parameters," George C. Marshall Space Flight Center, Huntsville, Ala., Memo R-ASTR-NGA, Sept. 1968.

[3] Hendley, A. C., "Quaternions for Control of Space Vehicles," *Proceedings of the Institute of Navigation, National Space Meeting on Space Shuttle—Space Station—Nuclear Shuttle Navigation*, Huntsville, Ala., Feb. 1971.

Lubricant Reservoir Systems: Thermal Considerations

L. M. Dormant* and S. Feuerstein†

The Aerospace Corporation, El Segundo, Calif.

Nomenclature

A, B	=	constants for a liquid, Eq. (1)
H	=	molar heat of vaporization of liquid
M	=	molecular weight
p	=	pressure
R	=	universal gas constant
r	=	nylon pore radius
T	=	temperature, K
t	=	time
$\bar{V}$	=	molar volume of liquid (molecular weight/density)
w	=	weight loss per unit area
γ	=	surface tension of liquid
ρ	=	density

Subscripts

p_0	=	vapor pressure of bulk liquid at given temperature, Eq. (2)
T_B	=	chamber B temperature
T_C	=	chamber C temperature

Introduction

In an earlier paper,[1] it was determined that, under isothermal conditions, porous nylon blocks used as lubricant replenishment reservoirs in such spacecraft components as despun mechanical assemblies actually behave as lubricant sinks or sponges. The cause of such behavior is capillary effects. In this paper, the thermal conditions necessary to insure that a porous reservoir functions satisfactorily as a lubricant replenishment source are derived by thermodynamic methodology.

Several active feed types of reservoir systems have been proposed by spacecraft contractors. The primary problem is knowing the amount of lubricant to be released and the rate of its release. Computations of these quantities have been based only on the amount of lubricant lost by vapor phase trans-

Received April 16, 1976; revision received June 1, 1976. This work was supported by the U.S. Air Force under Contract No. F0470-75-C-0076.

Index categories: Spacecraft Communication Systems; Liquid and Solid Thermophysical Properties.

*Formerly Member of the Technical Staff, Chemistry and Physics Laboratory.

†Head, Interfacial Science Department, Chemistry and Physics Laboratory.