[3]Tao, G., and Kokotovic, P. V., "Continuous-Time Adaptive Control of Systems with Unknown Backlash," *IEEE Transactions on Automatic Control*, Vol. 40, No. 6, 1995, pp. 1083–1087.

[4]Bernstein, D. S., and Haddad, W. M., "Nonlinear Controller for Positive Real Systems with Arbitrary Input Nonlinearities," *IEEE Transactions on Automatic Control*, Vol. 39, No. 7, 1994, pp. 1513–1517.

[5]Haddad, W. M., and Kapila, V., "Antiwindup Controllers for Systems with Input Nonlinearities," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1387–1390.

[6]Utkin, V. I., and Young, K. D., "Methods for Constructing Discontinuous Planes in Multidimensional Variable Structure Systems," *Automatic Remote Control*, Vol. 9, No. 10, 1978, pp. 1466–1470.

[7]Shyu, K. K., and Liu, C. Y., "Variable Structure Controller Design for Robust Tracking and Model Following," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 6, 1996, pp. 1395–1397.

Quaternion Parameterization and a Simple Algorithm for Global Attitude Estimation

R. G. Reynolds*
TRW Space and Electronics Group, Chantilly, Virginia 20151

Introduction

OF the many representations that exist for attitude, quaternions are popular for several reasons.

1) The propagation differential equations have no singularities, in contrast to, e.g., Euler's angles or Gibbs' vectors,[1] which require special handling near these points.

2) Modulo a scaling factor, the quaternion is a minimal three-degree-of-freedom representation of attitude as opposed to, e.g., direction cosine matrices, which must be orthonormalized in practical applications.

3) There are a number of simple and direct methods available to implement controllers with quaternion feedback, which are globally stable.[2–4]

In this Note, we examine the decomposition of the quaternion attitude representation in terms of a basis formed using observation unit vectors and a number of direct applications of this result. In particular, we will consider the implementation of a global attitude estimation scheme in both recursive and batch form. The latter batch algorithm may be shown to be identical to the quaternion estimation (QUEST) algorithm.[5]

Representations and Conventions

Vector arrays should always be considered to be column arrays, with superscript T indicating transposition to a row vector. We define the quaternion $\boldsymbol{q} \in \boldsymbol{H}$, the division ring of 4-tuple real quaternions,[6,7] with group composition given by

$$\boldsymbol{q}_{13} = \boldsymbol{q}_{12} \circ \boldsymbol{q}_{23} = \left(q_{12}Q_{23} + q_{23}Q_{12} + Q_{12} \times Q_{23},\ q_{12}q_{23} - Q_{12}^T Q_{23}\right) \tag{1}$$

componentwise addition, and identity $\vartheta = (0, 0, 0, 1)$. In Eq. (1), $\times$ denotes the vector cross product, and we have introduced the shorthand representation of a quaternion as $\boldsymbol{q} = (\boldsymbol{Q}, q)$, where $\boldsymbol{Q} \in \boldsymbol{R}^3$ is the so-called vector part of the quaternion and $q \in \boldsymbol{R}$ is the scalar part.

The subscript on a given quaternion in Eq. (1) is a helpful mnemonic, which indicates the rotation that it represents, e.g., $\boldsymbol{q}_{12}$ is associated with the rotation transformation from a given reference frame 1 to reference frame 2. Note that composition takes place in such a way that the common reference frame is contiguous.

There exist maps $p : V \to \boldsymbol{R}^3$ and $i : \boldsymbol{R}^3 \to V$ termed canonical projection and inclusion, respectively, which establish a correspondence between vectors in $\boldsymbol{R}^3$ and the subset $\boldsymbol{V} \subset \boldsymbol{H}$ consisting of quaternions with scalar part equal to zero. In general, the action of these operators shall be implicit. An example of this duality is the rotation of vectors in $\boldsymbol{R}^3$, which is identified with the conjugacy class of $\boldsymbol{V}$ in $\boldsymbol{H}$

$$V^* = \left\{ \boldsymbol{v}_1 = \boldsymbol{q}_{12} \circ \boldsymbol{v}_2 \circ \boldsymbol{q}_{12}^{-1} \,\middle|\, \boldsymbol{q}_{12} \in \boldsymbol{H}, \boldsymbol{v}_2 \in \boldsymbol{V} \right\} \tag{2}$$

Note how the subscripts line up in a consistent way in this representation.

For numerical considerations in practical applications, attention is often focused on the subgroup $\boldsymbol{U} \triangleleft \boldsymbol{H}$ of unit quaternions, those for which the Euclidean norm in $\boldsymbol{R}^4$ is unity, and for which inversion is trivial negation of the vector part of the quaternion. The unity magnitude of $\boldsymbol{q} \in \boldsymbol{U}$ implies $\boldsymbol{U} \sim S^3$, the three sphere endowed with the quaternion group structure. For clarity of presentation, quaternions in this Note are generally not constrained to be unit quaternions.

Quaternion Parameterization Using Observation Unit Vectors

Suppose that a unit vector $\boldsymbol{x}_1$ in a given Euclidean reference frame 1 is to be transformed to a new unit vector $\boldsymbol{x}_2$ in frame 2 via conjugation by the quaternion $\boldsymbol{q}_{12}$ as in Eq. (2). The quaternion in question might take the form of

$$\boldsymbol{q}_{12} \sim \left(\boldsymbol{x}_2 \times \boldsymbol{x}_1,\ 1 + \boldsymbol{x}_2^T \boldsymbol{x}_1\right) \tag{3}$$

where the $\sim$ symbol has been used to indicate that $\boldsymbol{q}_{12}$ is equivalent to any quaternion that may be obtained by scaling expression (3). This quaternion represents the minimum angle rotation required to transform $\boldsymbol{x}_1$ to $\boldsymbol{x}_2$.

If we visualize the transformation process, it is the rotation about an axis orthogonal to both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, as pictured in the top-center position in Fig. 1. Figure 1 is somewhat misleading, in that the vector part of the quaternion appears to be the negative axis of rotation. However, the quaternion represents the rotation of a frame of reference in which vector $\boldsymbol{x}_1$ appears as vector $\boldsymbol{x}_2$ following the rotation, and so the vector part of the quaternion is actually the positive axis of rotation in this sense.

Another quaternion that would rotate $\boldsymbol{x}_1$ into $\boldsymbol{x}_2$ is given by

$$\boldsymbol{q}_{12} \sim (\boldsymbol{x}_2 + \boldsymbol{x}_1, 0) \tag{4}$$

which represents a 180-deg rotation as shown in the lower-left position in Fig. 1.

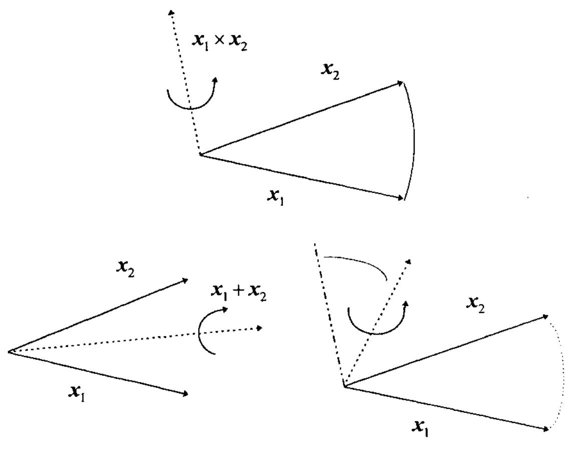

Fig. 1 Transformations.

Received June 3, 1997; revision received Feb. 27, 1998; accepted for publication March 8, 1998.

*Staff Engineer, Spacecraft Operations East, 14320 Sullyfield Circle, Suite 2100.

These two quaternions may be combined to create a parameterization of the set of quaternions that rotates x_1 into x_2

$$q_{12} \sim \left[(x_2 \times x_1)\cos(\phi/2) + (x_2 + x_1)\sin(\phi/2),\right.$$
$$\left.\left(1 + x_2^T x_1\right)\cos(\phi/2)\right] \tag{5}$$

so that the total rotation is as shown in the lower-right position in Fig. 1.

Another way to arrive at Eq. (5) is to combine the rotation of Eq. (3) with a twist about x_2

$$q_{12} \sim \left(x_2 \times x_1, 1 + x_2^T x_1\right) \circ [x_2 \sin(\phi/2), \cos(\phi/2)] \tag{6}$$

which demonstrates that Eq. (5) is, indeed, a full parameterization of feasible quaternions. Similarly, we may parameterize the entire set of quaternions that rotates another unit vector y_1 into y_2 by

$$q_{12} \sim \left[(y_2 \times y_1)\cos(\psi/2) + (y_2 + y_1)\sin(\psi/2),\right.$$
$$\left.\left(1 + y_2^T y_1\right)\cos(\psi/2)\right] \tag{7}$$

in terms of the angle ψ.

Now suppose we wish to determine the unique quaternion (modulo a scaling factor) that transforms x_1 to x_2 and y_1 to y_2. One way to do this would be to normalize and equate Eqs. (5) and (7), solve for ϕ or ψ, and plug the result back into the appropriate expression. However, a quicker way is to note that the vector part of expression (5) lies in a plane orthogonal to $x_2 - x_1$. Similarly, the vector part of Eq. (7) lies in the plane orthogonal to $y_2 - y_1$. Thus, if the planes intersect along a particular direction, the vector part of the quaternion we seek must lie here, i.e., in the direction $(x_2 - x_1) \times (y_2 - y_1)$.

The fourth element of the quaternion can be determined by ratioing the vector part to scalar part of the quaternion in Eq. (5) and equating this to the ratio of the vector part to the unknown fourth element and solving. The result is

$$q_{12} \sim \left[(x_2 - x_1) \times (y_2 - y_1), x_1^T y_2 - x_2^T y_1\right] \tag{8}$$

This construction breaks down in three instances.

1) If the vectors x_1 and y_1, hence, x_2 and y_2, are parallel, we do not have enough information to construct a unique quaternion.

2) If $x_1 = x_2$ and $y_1 = y_2$, then the quaternion is the identity.

3) If $x_2 - x_1$ is otherwise parallel to $y_2 - y_1$, then the planes defined by these normals coincide. Substitute in the normalized cross products $z_1 = (x_1 \times y_1)/\|x_1 \times y_1\|$ and $z_2 = (x_2 \times y_2)/\|x_2 \times y_2\|$ for either x_1 and x_2 or y_1 and y_2, and repeat the computation. If either vector $x_2 - x_1$ or $y_2 - y_1$ is identically zero, substitute z_2 and z_1 for the associated pair x_1 and x_2, or y_1 and y_2.

The rationale for these three conditions is that, assuming point 1 does not hold, the three vectors $x_2 - x_1$, $y_2 - y_1$, and $z_2 - z_1$ form a basis for the eigenvectors of the matrix $T_1^2 \in SO(3)$, the orthogonal rotation matrix[1] that corresponds to q_{12}, minus identity, i.e.,

$$[x_2 - x_1 \quad y_2 - y_1 \quad z_2 - z_1] = \left(T_1^2 - I\right)[x_1 \quad y_1 \quad z_1] \tag{9}$$

Point 2 covers the only case in which there is more than one invariant axis and, as a result, at least two of the cross products $(x_2 - x_1) \times (y_2 - y_1)$, $(z_2 - z_1) \times (x_2 - x_1)$, and $(y_2 - y_1) \times (z_2 - z_1)$ must be nonzero and parallel to the invariant rotation axis of the transformation.

Application: Direction Cosine Matrix to Quaternion

The first application of the quaternion decomposition that we consider is initialization of the quaternion from a given direction cosine matrix. Suppose this is given as $T_1^2 = [a \; b \; c]$, where a, b, and c are column three vectors that represent the transformation of e_1, e_2, and e_3, the canonical basis vectors, from frame 1 to frame 2. The quaternion is given by any of the following three parameterizations that are not trivially zero:

$$\begin{aligned} q_{12} &\sim [(a - e_1) \times (b - e_2), b_1 - a_2] \\ q_{12} &\sim [(a - e_1) \times (c - e_3), c_1 - a_3] \\ q_{12} &\sim [(b - e_2) \times (c - e_3), c_2 - b_3] \end{aligned} \tag{10}$$

Application: Global Attitude Estimation

Suppose that we have a set of N observation unit vectors s_i in a spacecraft body frame of reference, with corresponding unit vectors r_i in some reference frame of reference. Let us assume that both frames are inertially fixed for the duration of observation.

Equations (3) and (4) show that the ith set of vectors establishes a two-dimensional orthonormal basis (in the quaternion algebra) for the quaternion from reference to spacecraft q_{rs}, i.e.,

$$b_{1_i} = \frac{\left(s_i \times r_i, 1 + s_i^T r_i\right)}{\sqrt{2\left(1 + s_i^T r_i\right)}}, \qquad b_{2_i} = \frac{(s_i + r_i, 0)}{\sqrt{2\left(1 + s_i^T r_i\right)}} \tag{11}$$

Similarly, there is a two-dimensional orthonormal basis for the kernel or null space of the quaternion covector q_{rs}^T

$$n_{1_i} = \frac{\left(r_i \times s_i, 1 - s_i^T r_i\right)}{\sqrt{2\left(1 - s_i^T r_i\right)}}, \qquad n_{2_i} = \frac{(s_i - r_i, 0)}{\sqrt{2\left(1 - s_i^T r_i\right)}} \tag{12}$$

Clearly, errors in the observations will cause these basis and null vectors to be somewhat in error so that, to estimate attitude, we would want to put them in some sort of least-squares estimation algorithm. One approach for this would be to seek to minimize the cost functional

$$J(q_{rs}) = \sum_{\substack{i=1,N \\ j=1,2}} \alpha_i \left(n_{ji}^T q_{rs}\right)^2 \tag{13}$$

with weighting coefficients α_i, subject to the constraint that q_{rs} have unit length. Another approach would seek to maximize the cost

$$J(q_{rs}) = \sum_{\substack{i=1,N \\ j=1,2}} \alpha_i \left(b_{ji}^T q_{rs}\right)^2 \tag{14}$$

subject to the same constraint.

Still a third approach would be to combine these two and seek to maximize

$$J(q_{rs}) = \sum_{\substack{i=1,N \\ j=1,2}} \alpha_i \left[\left(b_{ji}^T q_{rs}\right)^2 - \left(n_{ji}^T q_{rs}\right)^2\right] \tag{15}$$

or minimize the negative of this, whichever is preferred. It turns out that this cost function is identical to that originally proposed by Wahba[8] for least-squares quaternion estimation.

The addition of the constraint that q_{rs} be a unit quaternion results in an eigenvalue problem, with q_{rs} being the eigenvector associated with the maximum eigenvalue of

$$K = \sum_{\substack{i=1,N \\ j=1,2}} \alpha_i \left[b_{ji} b_{ji}^T - n_{ji} n_{ji}^T\right] \tag{16}$$

This is the q method originally devised by Davenport[1] to minimize Wahba's cost functional, which was later refined by Shuster,[5] and an efficient solution to this problem is his QUEST algorithm, which has become well known and frequently utilized in the spacecraft attitude determination community.

Suppose we do not have the computing resources to implement this batch algorithm or, alternatively, suppose that our spacecraft frame of reference is really not inertial, and we wish to track the variation of the quaternion over time. A recursive algorithm is then desirable.

The existence and knowledge of the two null vectors given suggests an extremely simple algorithm based on Gram–Schmidt orthogonalization, i.e., if the null vectors were known perfectly, we might take an initial estimate of the quaternion and orthogonalize it with respect to them:

$$q_{rs} \to q_{rs} - \left(n_1^T q_{rs}\right)n_1 - \left(n_2^T q_{rs}\right)n_2 \tag{17}$$

followed by normalization to maintain good numerical properties of the quaternion. The algorithm can be equipped with some small gain factor, and we have the following theorem.

Theorem: Given a unit quaternion $\boldsymbol{q}$ and an infinite sequence of quaternion null vectors $\{\boldsymbol{n}_i\}$ in which any infinite subsequence spans $\aleph\{\boldsymbol{q}^T\}$, the kernel of the quaternion covector $\boldsymbol{q}^T$ and an initial unit quaternion estimate $\boldsymbol{q}_0$ that does not lie entirely in $\aleph\{\boldsymbol{q}^T\}$, the recursion

$$\boldsymbol{q}_{i+1} = \frac{\boldsymbol{q}_i - \alpha\left(\boldsymbol{n}_i^T\boldsymbol{q}_i\right)\boldsymbol{n}_i}{\sqrt{1-\alpha(2-\alpha)\left(\boldsymbol{n}_i^T\boldsymbol{q}_i\right)^2}} \tag{18}$$

for some $\alpha \in (0, 1]$ is globally convergent to the true quaternion $\boldsymbol{q}$.

Proof: Let $\boldsymbol{M} = \boldsymbol{M}^2 = \boldsymbol{M}^T$ be a nonnegative definite symmetric projection matrix whose kernel spans the image of $\boldsymbol{q}$. The function

$$V_i = \boldsymbol{q}_i^T\boldsymbol{M}\boldsymbol{q}_i \tag{19}$$

is a radially unbounded Lyapunov function for which

$$\Delta V_i = V_{i+1} - V_i = \frac{\alpha(2-\alpha)\left(\boldsymbol{n}_i^T\boldsymbol{q}_i\right)^2}{1-\alpha(2-\alpha)\left(\boldsymbol{n}_i^T\boldsymbol{q}_i\right)^2}(V_i - 1) \tag{20}$$

demonstrating that V_i is monotonically decreasing except in two cases: 1) when $\boldsymbol{n}_i^T\boldsymbol{q}_i = 0$, which cannot persist unless $\boldsymbol{q}_i = \boldsymbol{q}$ as desired, and 2) when $V_i = 1$, which undermines our requirement that $\boldsymbol{q}_0$ have at least some component in $\mathrm{Im}\{\boldsymbol{q}\}$. Thus, $V_i \to 0 \Rightarrow \boldsymbol{q}_i \to \boldsymbol{q}$.

To deal with imperfect data, we can specify α as small as we like. In general, an optimal gain factor, in terms of attitude uncertainty, will be the ratio of desired attitude variance, divided by the sum of attitude variance and measurement variance; however, the details will be left to the reader.

One weakness in the preceding algorithm may be readily observed in that the null vectors computed according to Eq. (12) are undefined when $\boldsymbol{r}_i \equiv \boldsymbol{s}_i$. However, because the basis vectors are still defined, we can use the fact that

$$\boldsymbol{n}_{1i}\boldsymbol{n}_{1i}^T + \boldsymbol{n}_{2i}\boldsymbol{n}_{2i}^T + \boldsymbol{b}_{1i}\boldsymbol{b}_{1i}^T + \boldsymbol{b}_{2i}\boldsymbol{b}_{2i}^T = \boldsymbol{I} \tag{21}$$

to construct a projection matrix $\boldsymbol{P}_i = \boldsymbol{n}_{1i}\boldsymbol{n}_{1i}^T + \boldsymbol{n}_{2i}\boldsymbol{n}_{2i}^T = \boldsymbol{I} - \boldsymbol{b}_{1i}\boldsymbol{b}_{1i}^T - \boldsymbol{b}_{2i}\boldsymbol{b}_{2i}^T$ and an equivalent algorithm

$$\boldsymbol{q}_{i+1} = \frac{(\boldsymbol{I} - \alpha\boldsymbol{P}_i)\boldsymbol{q}_i}{\sqrt{1-\alpha(2-\alpha)\boldsymbol{q}_i^T\boldsymbol{P}_i\boldsymbol{q}_i}} \tag{22}$$

It is not difficult to show that

$$\boldsymbol{P}_i = \frac{1}{2}\begin{bmatrix} \left(1+\boldsymbol{s}_i^T\boldsymbol{r}_i\right)\boldsymbol{I} - \left(\boldsymbol{r}_i\boldsymbol{s}_i^T + \boldsymbol{s}_i\boldsymbol{r}_i^T\right) & \boldsymbol{r}_i \times \boldsymbol{s}_i \\ (\boldsymbol{r}_i \times \boldsymbol{s}_i)^T & 1 - \boldsymbol{s}_i^T\boldsymbol{r}_i \end{bmatrix} \tag{23}$$

In concluding this discussion of attitude determination, we assert that the same analysis can be used to verify convergence where the quaternion is dynamically changing in time. That is, if the projection of the quaternion forward in time is accomplished by

$$\boldsymbol{q}_{i+1} = \boldsymbol{A}_i\boldsymbol{q}_i \tag{24}$$

where $\boldsymbol{A}_i$ is a unitary operator (inverse is equal to the transpose) then, because $\aleph(\boldsymbol{q}_{i+1}^T) = \boldsymbol{A}_i\aleph(\boldsymbol{q}_i^T)$, any future null vectors can be projected back to an arbitrary point in time, and the analysis will show that convergence is achieved at that time, hence at all future times.

Conclusions

The outlined quaternion decomposition reveals previously hidden structure and nuance in the quaternion attitude representation. In addition, it opens the way to some exceedingly simple and direct methods of accomplishing various attitude determination and control tasks.

In particular, the batch attitude determination algorithm was shown to be equivalent to the QUEST algorithm, while revealing the actual mechanism by which the algorithm accomplishes its objective. The recursive algorithm given is simple, is globally convergent, and may be optimized to some extent. Further optimization of the algorithm could be accomplished by including a projection matrix or Kalman filter type covariance matrix to weight succeeding measurements based on previous updates; however, the algorithm as given is adequate for a wide variety of applications.

Acknowledgment

The author wishes to thank F. L. Markley of Goddard Space Flight Center for pointing out the resolution of the singularity problem in the attitude update equation via cancellation in the full projection matrix and for providing some of the history of the Wahba-inspired least-squares quaternion estimation effort.

References

[1]Wertz, J. R., *Spacecraft Attitude Determination and Control*, Kluwer Academic, Boston, 1979, Chap. 12.

[2]Joshi, S. M., Kelkar, A. G., and Wen, J. T.-Y.,"Robust Attitude Stabilization of Spacecraft Using Nonlinear Quaternion Feedback," *IEEE Transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp.1800–1803.

[3]Lizarralde, F., and Wen, J. T., "Attitude Control Without Angular Velocity Measurement: A Passivity Approach," *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, 1996, pp. 468–472.

[4]Wen, J. T.-Y., and Kreutz-Delgado, K., "The Attitude Control Problem," *IEEE Transactions on Automatic Control*, Vol. 36, No. 10, 1991, pp. 1148–1162.

[5]Shuster, M. D., and Oh, S. D., "Three-Axis Attitude Determination from Vector Observations," *Journal of Guidance and Control*, Vol. 4, No. 1, 1981.

[6]Dummit, D. S., and Foote, R. M., *Abstract Algebra*, Prentice–Hall, Englewood Cliffs, NJ, 1991, pp. 34 and 226.

[7]Hungerford, T. E., *Algebra*, Springer–Verlag, New York, 1974, pp. 33 and 97.

[8]Wahba, G., "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, Vol. 7, No. 3, p. 409, Problem 65-1.