The superiority of MJC is maintained for wing-opening durations of $0.05-0.15$, although for wing-deployment periods larger than 0.2, the advantage of MJC over SDH is diminished. Because of uncertainties in the wing deployment mechanism and because larger deployment periods may require more complex mechanisms, durations less than 0.2 s should not be ignored. This implies that MJC appears to be useful for the described application.

V. Conclusions

The synthesis problem of wing deployment of a UAV was considered via the jump Markovian systems approach. The change in the vehicle's dynamics due to wing deployment was given a probabilistic interpretation, where the probability of deploying the wings from a folded position increases exponentially with time. However, once deployed, the wings remain open and, therefore, have a zero probability of switching back to a folded position. This description may even be closer to reality than the more intuitive deterministic one whereby the instant of wing deployment is selected according to a set of a priori defined conditions on the angle attack and its derivative, for example, small absolute value of the angle of attack combined with a small absolute derivative of the angle of attack. Even in the deterministic case, the first moment during which the described conditions are satisfied may depend on random parameters such as the angle of attack of the host vehicle at release, its velocity, etc., giving rise, in a very natural way, to a probabilistic interpretation of the switching time. For the situation where wing deployment timing is defined a priori, the transition probabilities may be considered to be design parameters tuned to maximizing the closed-loop bandwidth, minimizing control effort, etc.

The results achieved by the Markovian jump systems approach are quite encouraging. For very short-duration wing deployment, the Markov-jump-theory-based disturbance attenuation factor is markedly less than those obtained with the more common approaches of simply treating each wing state separately or requiring quadratic stability. The latter uses a single gain matrix to control the plant during its two different phases and, therefore, does not utilize the information about the parameter jumps, thus resulting in poor performance with a large control effort. Separately treating the closed- and open-wing systems does utilize the jump information, but does not account for the transient, thus leading to good disturbance attenuation, but at the cost of large controls. The Markovian jump approach uses different gain vectors for each phase while accounting for the jump. Although the probabilistic modeling of the jump may seem somewhat artificial in the case of a single-shot wing deployment operation, where no folding back of the wings is possible, the approach suggested can be thought of as a gain scheduling approach for systems with discrete operating points. This approach loses its advantage when the transition between these operating points is slow. The suggested procedure of comparing performance levels using both the deterministic and stochastic frameworks may be a useful addition to the overall control design process.

Although the application presented for the Markov jump approach, wing deployment in a UAV, is not very common in the aerospace industry, the design, analysis and simulation procedure suggested here may be relevant for other applications, where possibly fast enough transitions occur between discrete operating points.

Acknowledgment

The authors are indebted to Aron W. Pila for reading the manuscript, making numerous remarks, and improving it both in form and content.

References

[1] Gahinet, P., "Explicit Controller Formulas for LMI-Based H_∞ Synthesis," *Automatica*, Vol. 32, No. 7, 1996, pp. 1007–1014.

[2] Dragan, V., and Morozan, T., "Stability and Robust Stabilization to Linear Stochastic Systems Described by Differential Equations with Markovian Jumping and Multiplicative White Noise," Preprint Series, Inst. of Mathematics of the Romanian Academy, 1999, No. 17/1999.

[3] deFarias, D. P., Geromel, J. C., doVal, J. B. R., and Costa, O. L. V., "Output Feedback Control of Jump Markov Linear Systems in Continuous Time," *Transactions on Automatic Control*, Vol. AC-45, 2000, pp. 944–948.

[4] McRuer, D., Ashkenas, I., and Graham, D., *Aircraft Dynamics and Automatic Control*, Princeton Univ. Press, Princeton, NJ, 1973.

[5] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequality in Systems and Control Theory*, Frontier Series, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[6] Gahinet, P., Nemirovsky, A., Laub, A. J., and Chilali, M., "LMI Control Toolbox for Use with MATLAB," Mathworks, Inc., Natick, MA, 1995.

Fast Quaternion Attitude Estimation from Two Vector Measurements

F. Landis Markley*
NASA Goddard Space Flight Center, Greenbelt, Maryland 20771

Introduction

MANY spacecraft attitude-determination methods use exactly two vector measurements, e.g., unit vectors along the line of sight to a star or the sun or along the Earth's magnetic field. We want to find the attitude matrix that transforms vectors from some reference frame to the spacecraft body frame. That is, we would like to find a 3×3 proper orthogonal matrix A such that

$$A\boldsymbol{r}_i = \boldsymbol{b}_i \qquad \text{for} \qquad i = 1, 2 \tag{1}$$

where $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the measured unit vectors in the spacecraft body frame and $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the corresponding unit vectors in the reference frame. It is impossible to satisfy both of these equations in general because they imply that $\boldsymbol{b}_1\cdot\boldsymbol{b}_2=\boldsymbol{r}_1\cdot\boldsymbol{r}_2$, which might not be true in the presence of measurement errors. All reasonable two-vector attitude-determination schemes give the same estimate when this equality holds, however.

The earliest algorithm for determining spacecraft attitude from two vector measurements was the TRIAD algorithm,[1,2] which is simple to implement but does not treat the observations optimally. Wahba[3] proposed that the optimal attitude matrix should minimize the loss function[4]

$$L(A) \equiv \frac{1}{2}\sum_i a_i|\boldsymbol{b}_i - A\boldsymbol{r}_i|^2 = \sum_i a_i - \mathrm{trace}(AB^T) \tag{2}$$

where the a_i are positive weights assigned to the measurements and

$$B \equiv \sum_i a_i \boldsymbol{b}_i \boldsymbol{r}_i^T \tag{3}$$

Shuster showed a simplification of his optimal QUEST algorithm for the two-observation Wahba problem,[5] but Ref. 6 presented the first explicit closed-form optimal solution. Most existing optimal two-observation algorithms are significantly slower than TRIAD and actually slower than optimal n-observation algorithms. Recent exceptions are Mortari's optimal EULER-2 algorithm[7] and a suboptimal algorithm proposed by Reynolds.[8] The present Note presents two new algorithms for quaternion estimation from two vector measurements. The first is a very efficient optimal algorithm, which is almost as fast as the TRIAD algorithm. The second produces the same suboptimal estimate as TRIAD, but at reduced computational cost.

Received 11 June 2001; revision received 3 October 2001; accepted for publication 5 October 2001.

*Aerospace Engineer, Guidance, Navigation, and Control Systems Engineering Branch, Code 571. Fellow AIAA.

Optimal Two-Observation Quaternion Estimation Method

Following Davenport, we use the quaternion parameterization of the attitude to write the loss function as[2,4]

$$L(A) = \sum_i a_i - \begin{bmatrix} \boldsymbol{q} \\ q_4 \end{bmatrix}^T \begin{bmatrix} B + B^T - I\,\mathrm{tr}B & \sum_i a_i \boldsymbol{b}_i \times \boldsymbol{r}_i \\ \sum_i a_i (\boldsymbol{b}_i \times \boldsymbol{r}_i)^T & \mathrm{tr}B \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ q_4 \end{bmatrix} \tag{4}$$

where $\boldsymbol{q}$ and q_4 are the vector and scalar parts of the quaternion, respectively. We follow Shuster's quaternion conventions, but our treatment can be easily adapted to other conventions.[9–11]

In the two-observation case it is useful to define the normalized cross products

$$\boldsymbol{r}_3 \equiv (\boldsymbol{r}_1 \times \boldsymbol{r}_2)/|\boldsymbol{r}_1 \times \boldsymbol{r}_2|, \qquad \boldsymbol{b}_3 \equiv (\boldsymbol{b}_1 \times \boldsymbol{b}_2)/|\boldsymbol{b}_1 \times \boldsymbol{b}_2| \tag{5}$$

We note that $\boldsymbol{r}_3$ or $\boldsymbol{b}_3$ is undefined if the reference vectors or the observed vectors, respectively, are collinear, which means that there is insufficient information to determine the attitude uniquely. The optimal estimate must result in $A_{\mathrm{opt}}\boldsymbol{r}_1$ and $A_{\mathrm{opt}}\boldsymbol{r}_2$ being coplanar with $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, which means that the optimal transformation maps the cross-product vectors as

$$A_{\mathrm{opt}}\boldsymbol{r}_3 = \boldsymbol{b}_3 \tag{6}$$

This can be seen from the explicit solution,[6] geometrical reasoning,[7] or simply observing that any out-of-plane component of $A_{\mathrm{opt}}\boldsymbol{r}_1$ or $A_{\mathrm{opt}}\boldsymbol{r}_2$ would increase Wahba's loss function. The TRIAD estimate, although not optimal, always obeys Eq. (6), but not all two-measurement estimates do.[12]

Reynolds[8] observed that the most general transformation taking $\boldsymbol{r}_3$ into $\boldsymbol{b}_3$ can be expressed as a rotation through an arbitrary angle ϕ_r about $\boldsymbol{r}_3$ followed by the transformation that takes $\boldsymbol{r}_3$ into $\boldsymbol{b}_3$ using the minimum-angle rotation and then by a rotation through an arbitrary angle ϕ_b about $\boldsymbol{b}_3$. This has the quaternion representation, up to an irrelevant overall sign,

$$\begin{aligned} q(\phi) &= \begin{bmatrix} \boldsymbol{b}_3 \sin(\phi_b/2) \\ \cos(\phi_b/2) \end{bmatrix} \otimes q_{\min} \otimes \begin{bmatrix} \boldsymbol{r}_3 \sin(\phi_r/2) \\ \cos(\phi_r/2) \end{bmatrix} \\ &= q_{\min}\cos(\phi/2) + q_{180}\sin(\phi/2) \end{aligned} \tag{7}$$

where

$$q_{\min} \equiv \frac{1}{\sqrt{2(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)}} \begin{bmatrix} \boldsymbol{b}_3 \times \boldsymbol{r}_3 \\ 1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3 \end{bmatrix} \tag{8}$$

$$q_{180} \equiv \frac{1}{\sqrt{2(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)}} \begin{bmatrix} \boldsymbol{b}_3 + \boldsymbol{r}_3 \\ 0 \end{bmatrix} \tag{9}$$

and $\phi \equiv \phi_b + \phi_r$. The quaternion q_{180} transforms $\boldsymbol{r}_3$ into $\boldsymbol{b}_3$ by means of a 180-deg rotation about the bisector of these two vectors. We find the optimal quaternion by inserting Eq. (7) into Eq. (4) and finding the angle ϕ that minimizes the loss function. The half-angle formulas of trigonometry and some vector algebra give

$$L(A) = a_1 + a_2 - (1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)^{-1}(\alpha\cos\phi + \beta\sin\phi) \tag{10}$$

where

$$\begin{aligned} \alpha \equiv{}& (1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)(a_1 \boldsymbol{b}_1 \cdot \boldsymbol{r}_1 + a_2 \boldsymbol{b}_2 \cdot \boldsymbol{r}_2) \\ &+ (\boldsymbol{b}_3 \times \boldsymbol{r}_3) \cdot (a_1 \boldsymbol{b}_1 \times \boldsymbol{r}_1 + a_2 \boldsymbol{b}_2 \times \boldsymbol{r}_2) \end{aligned} \tag{11}$$

$$\beta \equiv (\boldsymbol{b}_3 + \boldsymbol{r}_3) \cdot (a_1 \boldsymbol{b}_1 \times \boldsymbol{r}_1 + a_2 \boldsymbol{b}_2 \times \boldsymbol{r}_2) \tag{12}$$

The loss function is minimized by setting $\cos\phi = \alpha/\gamma$ and $\sin\phi = \beta/\gamma$ with

$$\gamma \equiv \sqrt{\alpha^2 + \beta^2} \tag{13}$$

which gives

$$L(A_{\mathrm{opt}}) = a_1 + a_2 - (1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)^{-1}\gamma \tag{14}$$

We avoid singularities for $\cos\phi = \pm 1$ by using alternative forms of trigonometric half-angle relations to write the optimal quaternion estimate as

$$q_{\mathrm{opt}} = \begin{cases} \dfrac{1}{2\sqrt{\gamma(\gamma+\alpha)(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)}} \begin{bmatrix} (\gamma+\alpha)(\boldsymbol{b}_3 \times \boldsymbol{r}_3) + \beta(\boldsymbol{b}_3 + \boldsymbol{r}_3) \\ (\gamma+\alpha)(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3) \end{bmatrix} & \text{for} \quad \alpha \ge 0 \\[2ex] \dfrac{1}{2\sqrt{\gamma(\gamma-\alpha)(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3)}} \begin{bmatrix} \beta(\boldsymbol{b}_3 \times \boldsymbol{r}_3) + (\gamma-\alpha)(\boldsymbol{b}_3 + \boldsymbol{r}_3) \\ \beta(1 + \boldsymbol{b}_3 \cdot \boldsymbol{r}_3) \end{bmatrix} & \text{for} \quad \alpha \le 0 \end{cases} \tag{15}$$

If $\alpha = 0$, $\gamma = |\beta|$ from Eq. (13) so that these equations are identical except possibly for an irrelevant overall sign.

The preceding expressions are indeterminate when $\boldsymbol{b}_3 = -\boldsymbol{r}_3$. This condition can be avoided by solving for the attitude with respect to a reference coordinate frame related to the original reference frame by a 180-deg rotation about one of the coordinate axes.[4,5,13] This rotation is easy to implement on the input data because a rotation about any coordinate axis simply changes the signs of the components of $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$ along the other two axes. The effect of a rotation about the ith axis on the inner product $\boldsymbol{b}_3 \cdot \boldsymbol{r}_3$ is thus

$$\begin{aligned} (\boldsymbol{b}_3 \cdot \boldsymbol{r}_3)_{\mathrm{rotated}} &= [(\boldsymbol{b}_3)_i(\boldsymbol{r}_3)_i - (\boldsymbol{b}_3)_j(\boldsymbol{r}_3)_j - (\boldsymbol{b}_3)_k(\boldsymbol{r}_3)_k]_{\mathrm{unrotated}} \\ &= [2(\boldsymbol{b}_3)_i(\boldsymbol{r}_3)_i - \boldsymbol{b}_3 \cdot \boldsymbol{r}_3]_{\mathrm{unrotated}} \end{aligned} \tag{16}$$

We ensure the largest value for $\boldsymbol{b}_3 \cdot \boldsymbol{r}_3$ in Eqs. (14) and (15) by performing the estimation in the original reference frame if $\boldsymbol{b}_3 \cdot \boldsymbol{r}_3$ is larger than any of the products $(\boldsymbol{b}_3)_i(\boldsymbol{r}_3)_i$ in this frame, or rotating about the axis with the maximum product $(\boldsymbol{b}_3)_i(\boldsymbol{r}_3)_i$ if this is larger than $\boldsymbol{b}_3 \cdot \boldsymbol{r}_3$. In the latter case the estimated quaternion q^i is easily restored to the original frame by

$$q \equiv q^i \otimes \begin{bmatrix} \boldsymbol{e}^i \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}^i \\ q_4^i \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{e}^i \\ 0 \end{bmatrix} = \begin{bmatrix} q_4^i \boldsymbol{e}^i - \boldsymbol{q}^i \times \boldsymbol{e}^i \\ -\boldsymbol{q}^i \cdot \boldsymbol{e}^i \end{bmatrix} \tag{17}$$

where $\boldsymbol{e}^i$ is the unit vector along the i^{th} coordinate axis. This requires only permutations and sign changes of the quaternion components.

Suboptimal Two-Observation Quaternion Estimation Method

The optimal estimate gives $A\boldsymbol{r}_1 = \boldsymbol{b}_1$ in the limit that the first measurement is assigned much more weight than the second. This is often the case of interest, as when combining measurements from a fine digital sun sensor with arc-minute accuracy and a triaxial magnetometer with errors of a few degrees, and is the case treated by TRIAD. The quaternion estimate taking $\boldsymbol{r}_1$ into $\boldsymbol{b}_1$ is given, in analogy with Eq. (7), by

$$\begin{aligned} q(\psi) = \frac{1}{\sqrt{2(1 + \boldsymbol{b}_1 \cdot \boldsymbol{r}_1)}} \Bigg\{ &\cos\left(\frac{\psi}{2}\right) \begin{bmatrix} \boldsymbol{b}_1 \times \boldsymbol{r}_1 \\ 1 + \boldsymbol{b}_1 \cdot \boldsymbol{r}_1 \end{bmatrix} \\ &+ \sin\left(\frac{\psi}{2}\right) \begin{bmatrix} \boldsymbol{b}_1 + \boldsymbol{r}_1 \\ 0 \end{bmatrix} \Bigg\} \end{aligned} \tag{18}$$

In parallel with the optimal case, we find

$$\begin{aligned} L(A) = a_2\big[&1 - (\boldsymbol{b}_1 \cdot \boldsymbol{b}_2)(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2) \\ &- (1 + \boldsymbol{b}_1 \cdot \boldsymbol{r}_1)^{-1}(\mu\cos\psi + \nu\sin\psi)\big] \end{aligned} \tag{19}$$

where

$$\mu \equiv (1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)[(\boldsymbol{b}_1\times\boldsymbol{b}_2)\cdot(\boldsymbol{r}_1\times\boldsymbol{r}_2)] - [\boldsymbol{b}_1\cdot(\boldsymbol{r}_1\times\boldsymbol{r}_2)][\boldsymbol{r}_1\cdot(\boldsymbol{b}_1\times\boldsymbol{b}_2)] \tag{20}$$

and

$$\nu \equiv (\boldsymbol{b}_1+\boldsymbol{r}_1)\cdot[(\boldsymbol{b}_1\times\boldsymbol{b}_2)\times(\boldsymbol{r}_1\times\boldsymbol{r}_2)] \tag{21}$$

The loss function is minimized by setting $\cos\psi = \mu/\rho$ and $\sin\psi = \nu/\rho$, with

$$\rho \equiv \sqrt{\mu^2+\nu^2} \tag{22}$$

which gives

$$L(A_{\mathrm{TRIAD}}) = a_2\big[1 - (\boldsymbol{b}_1\cdot\boldsymbol{b}_2)(\boldsymbol{r}_1\cdot\boldsymbol{r}_2) - (1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)^{-1}\rho\big] \tag{23}$$

and

$$q_{\mathrm{TRIAD}} = \begin{cases} \dfrac{1}{2\sqrt{\rho(\rho+\mu)(1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)}}\begin{bmatrix}(\rho+\mu)(\boldsymbol{b}_1\times\boldsymbol{r}_1)+\nu(\boldsymbol{b}_1+\boldsymbol{r}_1)\\ (\rho+\mu)(1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)\end{bmatrix} & \text{for} \quad \mu\ge 0\\[2ex] \dfrac{1}{2\sqrt{\rho(\rho-\mu)(1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)}}\begin{bmatrix}\nu(\boldsymbol{b}_1\times\boldsymbol{r}_1)+(\rho-\mu)(\boldsymbol{b}_1+\boldsymbol{r}_1)\\ \nu(1+\boldsymbol{b}_1\cdot\boldsymbol{r}_1)\end{bmatrix} & \text{for} \quad \mu\le 0\end{cases} \tag{24}$$

This estimate is suboptimal for general measurement weights, but it is less expensive to compute than the optimal estimate because it avoids normalizing the cross products $\boldsymbol{r}_1\times\boldsymbol{r}_2$ and $\boldsymbol{b}_1\times\boldsymbol{b}_2$. If either of these cross products is zero, indicating collinearity of the reference or body frame vectors, Eqs. (20) and (21) show that the estimate is undefined. The case of $\boldsymbol{b}_1 = -\boldsymbol{r}_1$ is handled by reference frame rotations in the same manner as for the optimal estimate.

Performance

The computational speed of the new optimal algorithm is 153 or 155 MATLAB® floating point operations (flops), depending on whether or not a reference frame rotation is required. This includes evaluation of the loss function, which provides a useful check for rejecting erroneous data. The optimal algorithm is almost as fast as the nonoptimal TRIAD algorithm (118 flops) followed by the extraction of a quaternion from the attitude matrix (20 or 22 flops). The suboptimal quaternion estimation algorithm requires only 104 or 106 flops to compute the TRIAD attitude estimate.

Simulations were performed with random attitudes uniformly distributed on the rotation group manifold[14] and random observation vectors independently and uniformly distributed on the unit sphere. The weights used in the optimal estimation algorithm were the inverse variances of Gaussian random errors that were added to the measurements. These variances are generally known a priori, but Ref. 4 discusses errors arising from mismodeled weights. The attitude estimation error was computed as the magnitude of the rotation angle between the true and estimated attitudes.[4]

The first of two simulation scenarios had equal measurement weights to show the largest difference between the optimal and suboptimal estimates. Measurement errors appropriate for a coarse sun sensor and a triaxial magnetometer were simulated with standard deviations of 2 deg per axis. Figure 1 exhibits the optimal and suboptimal estimation errors for 1000 test cases as functions of the angle between the two observed vectors. These errors have been multiplied by $|\boldsymbol{b}_1\times\boldsymbol{b}_2|$ to show that they are roughly inversely proportional to this geometrical factor and to facilitate plotting on a convenient vertical scale. To highlight the differences between the two estimators in this scenario, the probability distributions of $|\boldsymbol{b}_1\times\boldsymbol{b}_2|$ times the estimation errors in 10,000 simulations are plotted in Fig. 2. The products of $|\boldsymbol{b}_1\times\boldsymbol{b}_2|$ and the errors of the optimal and suboptimal estimates are less than 5.3 and 5.6 deg, respectively, at the 95% confidence level and less than 6.7 and 6.9 deg at the 99% confidence level.

The second scenario used measurement standard deviations of 1 arc minute per axis on the first vector and 2 deg on the second, which would be appropriate for measurements from a fine digital sun sensor and a triaxial magnetometer, respectively. The suboptimal estimation errors were indistinguishable from the optimal in a figure analogous to Fig. 1, as was to be expected for this unequal-weight scenario.

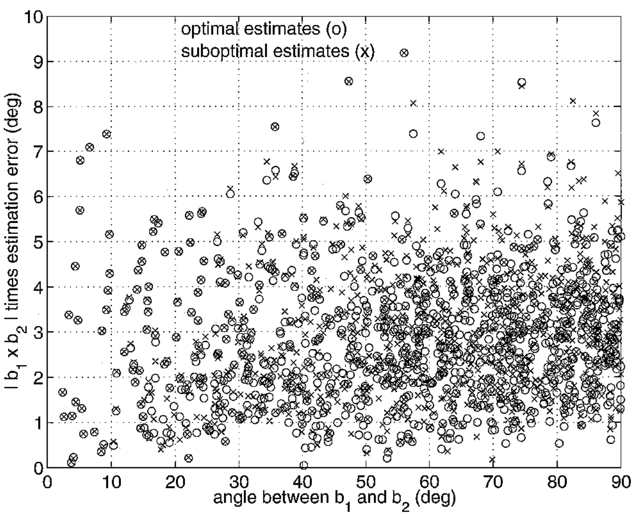

Fig. 1 Estimation errors of 1000 test cases with 2-deg measurement errors.

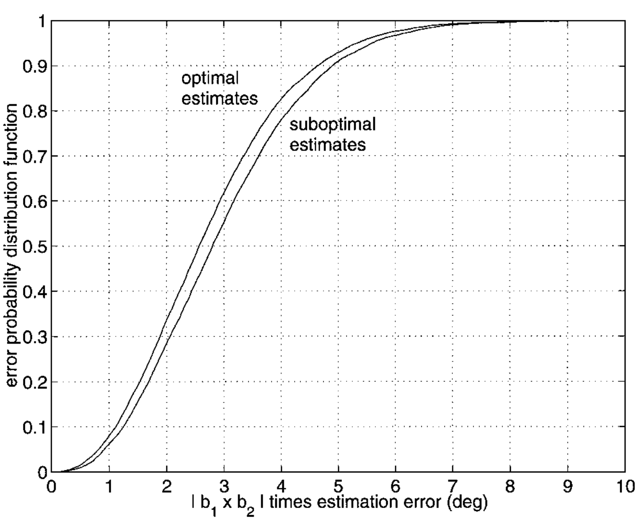

Fig. 2 Empirical estimation error probability distributions with 2-deg measurement errors.

Conclusions

We have presented two new algorithms for quaternion estimation using two vector measurements: an optimal algorithm that is almost as fast as the TRIAD algorithm and a suboptimal algorithm producing the TRIAD estimate at reduced computational cost. Tests show that the optimal estimates are marginally superior to the suboptimal estimates.

References

[1]Black, Harold D., "A Passive System for Determining the Attitude of a Satellite," *AIAA Journal*, Vol. 2, No. 7, 1964, pp. 1350, 1351.

[2]Lerner, Gerald M., "Three-Axis Attitude Determination," *Spacecraft Attitude Determination and Control*, edited by James R. Wertz, D. Reidel, Dordrecht, the Netherlands, 1978, pp. 420–428.

[3]Wahba, Grace, "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, Vol. 7, No. 3, 1965, p. 409.

[4]Markley, F. Landis, and Mortari, Daniele, "Quaternion Attitude Estimation Using Vector Observations," *Journal of the Astronautical Sciences*, Vol. 48, No. 2/3, 2000, pp. 359–380.

[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, pp. 70–77.

[6]Markley, F. Landis, "Attitude Determination Using Vector Observations: A Fast Optimal Matrix Algorithm," *Journal of the Astronautical Sciences*, Vol. 41, No. 2, 1993, pp. 261–280.

[7]Mortari, Daniele, "EULER-2 and EULER-n Algorithms for Attitude Determination from Vector Observations," *Space Technology*, Vol. 16, No. 5/6, 1996, pp. 317–321.

[8]Reynolds, R. G., "Quaternion Parameterization and a Simple Algorithm for Global Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 669–671.

[9]Markley, F. Landis, "Parameterization of the Attitude," *Spacecraft Attitude Determination and Control*, edited by James R. Wertz, D. Reidel, Dordrecht, the Netherlands, 1978, pp. 410–420.

[10]Shuster, Malcolm D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 439–517.

[11]Shepperd, S. W., "Quaternion from Rotation Matrix," *Journal of Guidance and Control*, Vol. 1, No. 3, 1978, pp. 223, 224.

[12]Markley, F. Landis, "Attitude Determination Using Two Vector Measurements," *1999 Flight Mechanics Symposium*, edited by John P. Lynch, NASA Goddard Space Flight Center, Greenbelt, MD, NASA/CP-1999-209235, 1999, pp. 39–52.

[13]Shuster, Malcolm D., and Natanson, Gregory A., "Quaternion Computation from a Geometric Point of View," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, 1993, pp. 545–556.

[14]Bauer, R., "Uniform Sampling of SO_3," *2001 Flight Mechanics Symposium*, edited by John P. Lynch, NASA Goddard Space Flight Center, Greenbelt, MD, NASA/CP-2001-209986, 2001, pp. 347–359.

Errata

Erratum for Improved Dispersion of a Fin-Stabilized Projectile Using a Passive Moveable Nose

Mark Costello and Raditya Agarwalla
Oregon State University, Corvallis, Oregon 97331

[J. Guidance, 23(5), pp. 900–903 (2000)]

THE six equations shown in the following contained typographical errors in the printed journal article. The corrected equations are given here:

$$A_F = -\frac{m_A m_F}{m_A + m_F} T_F S_{\rho F} \tag{9}$$

$$B_{\mathrm{AF}} = \frac{m_F}{m_A + m_F}\begin{Bmatrix} X_A \\ Y_A \\ Z_A \end{Bmatrix} - \frac{m_A}{m_A + m_F} T_F \begin{Bmatrix} X_F \\ Y_F \\ Z_F \end{Bmatrix} + \frac{m_A m_F}{m_A + m_F}\left(S_{\omega_A} S_{\omega_A} \begin{Bmatrix} \rho A_X \\ \rho A_Y \\ \rho A_Z \end{Bmatrix} - T_F S_{\omega_F} S_{\omega_F} \begin{Bmatrix} \rho F_X \\ \rho F_Y \\ \rho F_Z \end{Bmatrix} \right) \tag{10}$$

$$\begin{Bmatrix} L_C \\ M_C \\ N_C \end{Bmatrix} = \frac{M_g}{\sqrt{\sin^2(\theta_F) + \cos^2(\theta_F)\sin^2(\psi_F)}} \times \begin{Bmatrix} 0 \\ -\sin(\theta_F) \\ -\cos(\theta_F)\sin(\psi_F) \end{Bmatrix} + C_g \begin{Bmatrix} \delta p \\ \delta q \\ \delta r \end{Bmatrix} \tag{12}$$

$$T_F I_F \begin{Bmatrix} \dot p_F \\ \dot q_F \\ \dot r_F \end{Bmatrix} + T_F S_{\omega F} I_F \begin{Bmatrix} p_F \\ q_F \\ r_F \end{Bmatrix} = \begin{Bmatrix} L_C \\ M_C \\ N_C \end{Bmatrix} + T_F \begin{Bmatrix} L_F \\ M_F \\ N_F \end{Bmatrix} - S_{\tilde\rho_F} \begin{Bmatrix} X_C \\ Y_C \\ Z_C \end{Bmatrix} \tag{16}$$

$$\begin{bmatrix} I_A - S_{\rho_A} A_A & -S_{\rho_A} A_F \\ S_{\tilde\rho_F} A_A & T_F I_F + S_{\tilde\rho_F} A_F \end{bmatrix} \begin{Bmatrix} \dot p_A \\ \dot q_A \\ \dot r_A \\ \dot p_F \\ \dot q_F \\ \dot r_F \end{Bmatrix} = \begin{Bmatrix} g_{A_X} \\ g_{A_Y} \\ g_{A_Z} \\ g_{F_X} \\ g_{F_Y} \\ g_{F_Z} \end{Bmatrix} \tag{17}$$

$$\{g_F\} = -T_F S_{\omega_F} I_F \begin{Bmatrix} p_F \\ q_F \\ r_F \end{Bmatrix} + \begin{Bmatrix} L_C \\ M_C \\ N_C \end{Bmatrix} + T_F \begin{Bmatrix} L_F \\ M_F \\ N_F \end{Bmatrix} - S_{\tilde\rho_F}\{B_{\mathrm{AF}}\} \tag{19}$$

AIAA regrets the error.