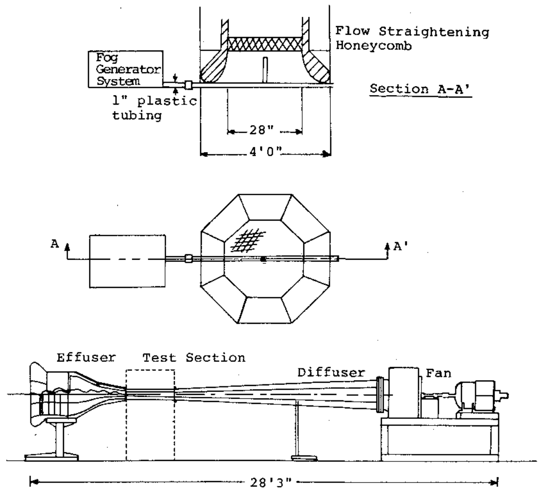

Fig. 3 Fog generator and subsonic wind tunnel.

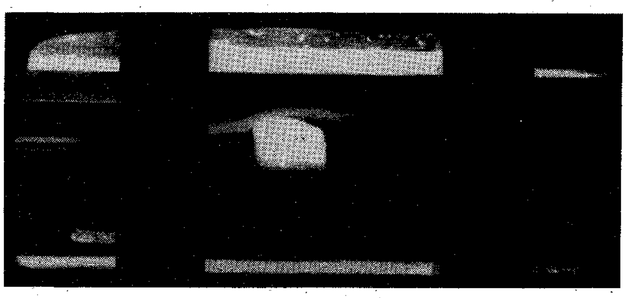

Fig. 4 Fog visualization photograph of magnetically suspended ring airfoil with separated flow at 85 fps (not spinning).

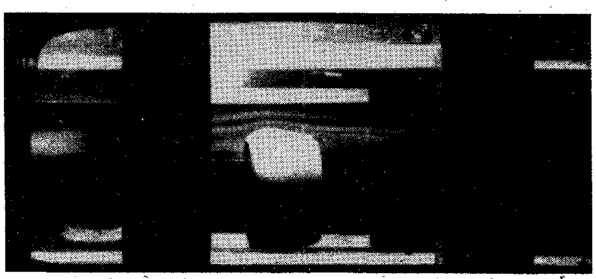

Fig. 5 Fog visualization photgraph of magnetically suspended ring airfoil with attached flow at 250 fps (not spinning).

The authors believe the water fog visualization technique also should be usable in atmospheric supersonic wind tunnels, if the total water flow rate is held sufficiently low to maintain the dewpoint in recirculation wind tunnels.

As suggested by a reviewer, other methods,[6] using higher vapor pressure substances, might be more useful for oscillatory, transient, or decelerating flows. It might be possible to use the present technique of liquid nitrogen mixing to quench the vapor phase of these other substances rapidly to produce a finer and denser smoke.

References

[1]Brown, F.N.M., "See the Wind Blow," University of Notre Dame, South Bend, Ind., 1971.

[2]Yu, J.P., Sparrow, E.M., and Eckert, E.R.G., "A Smoke Generator for Use in Fluid Flow Visualization," *International Journal of Heat Mass Transfer,* Vol. 15, 1972, pp.557–559.

[3]Smith, E., Reed, R.H., and Hodges, H.D., "The Measurement of Low Air Speeds by the Use of Titanium Tetrachloride," Texas Engineering Experimental Station, Research Rep. 25, May 1951.

[4]Maltby, R.L. and Keating, R.L.A., "Smoke Techniques for Use in Low Speed Wind Tunnels," *Flow Visualization in Tunnels using Indicators,* ADARDograph 70, pp. 87–109, 1962.

[5]Hodges, A.E. and Pount, T.N., "Further Development of a Smoke Producer using Vaporized Oil," Aerodynamics Note 233, Australian Defense Scientific Service Aeronautical Research Labs., Nov. 1964.

[6]Griffin, O.M. and Votaw, C.W., "The Use of Aerosols for the Visualization of Flow Phenomena," *International Journal of Heat and Mass Transfer,* Vol. 16, pp. 217–219.

[7]Vlajinac, M., "Design, Construction and Evaluation of a Subsonic Wind Tunnel," S.M. thesis, M.I.T., June 1970.

Computer Algorithms for Computation of Kinematical Relations for Three Attitude Angle Systems

Yoshiaki Ohkami*
National Aerospace Laboratory, Tokyo, Japan

Introduction

IN the study of attitude dynamics of spacecraft, it is commonplace to describe spacecraft attitude or orientation by a set of three attitude angles (typically Euler angles) through which a sequence of successive planar rotations is specified. In this description, a planar rotation is represented by a 3×3 orthogonal matrix, called a transformation matrix.[1] A general description of the orientation is generated by a single transformation matrix constructed as a matrix product of such matrices. In digital simulation of attitude dynamics,[2,3] a transformation matrix is often used together with another set of kinematical relations between angular velocities and the time derivatives of attitude angles (sometimes called Euler's kinematical equations). If a sequence of rotations is selected among many†, then the transformation matrix and angular velocity relations are uniquely determined.

Although the algebraic properties of such rotations have been studied extensively,[1,4] these kinematical relations are usually formed by manual calculation for a given ordered sequence and then programed on a digital computer. This is a laborious job especially when more than one coordinate systems must be employed.[2]

In this paper, a unified algorithm is developed for computation of the general transformation matrix on a digital computer ("general" in that if a sequence of any order is given, the transformation matrix is automatically generated without any manual calculation). An algorithm is also established for the general kinematical relations between angular velocities and the time derivatives of three attitude angles. In addition to these, algorithms for the inverse transformations are also developed, and thus a complete set of the kinematical relations is established for use in digital computer simulations. Particular emphasis is placed on the avoidance of unnecessary multiplications which otherwise cause excessive computation time.

Transformation Matrix

Consider the following coordinate systems; the first one (e.g., reference frame) is defined by the dextral (right-handed) orthogonal set of unit vectors (called a vector basis hereafter), $\{\boldsymbol{a}\}$, and the second one (e.g., body frame) by $\{\boldsymbol{d}\}$, and in addition to these, the two intermediate coordinate systems $\{\boldsymbol{b}\}$ and $\{\boldsymbol{c}\}$ which take place during a sequence of planar rotations.

Received May, 27, 1975; revision received March 31, 1976.
Index category: Spacecraft Attitude Dynamics and Control.

*Space Technology Research Group. Member AIAA.

†A simple combinatory calculation shows that there exist twelve meaningful sequences, four of which are essentially different from each other.

The sequence is defined as follows assuming the axes are labeled 1, 2, and 3 (as opposed to x, y, and z). First, rotate the first frame through an angle ϕ_1 about the r (1) axis of $\{a\}$ to produce the frame $\{b\}$; second, rotate the frame $\{b\}$ through an angle ϕ_2 about the r (2) axis of $\{b\}$ to produce $\{c\}$; and finally $\{c\}$ through ϕ_3 about the r (3) axis to obtain the frame $\{d\}$. If, for example, the second rotation is about the 3 axis, then $r(j)=r(2)=3$. Then the relations between the vector bases may be written in terms of matrices

$$\{b\}=C^{r(1)}(\phi_1)\cdot\{a\}, \qquad \{c\}=C^{r(2)}(\phi_2)\cdot\{b\}, \qquad \{d\}=C^{r(3)}(\phi_3)\cdot\{c\} \tag{1}$$

where $C^{r(i)}(\phi_i)$ is a 3×3 matrix representing the ith planar rotation through angle ϕ_i about the $r(i)$ axis. If the $\alpha\beta$ element of $C^{r(i)}(\phi_i)$ is denoted by $C^{r(i)}_{\alpha\beta}$ (dropping the argument ϕ_i without confusion), then the basic property of a planar rotation produces the following relations:

$$C^{r(i)}_{r(i),r(i)}=1 \tag{2a}$$

$$C^{r(i)}_{r(i),j}=C^{r(i)}_{j,r(i)}=0 \quad j\neq r(i) \tag{2b}$$

$$C^{r(i)}_{r(i)+1,r(i)+1}=C^{r(i)}_{r(i)+2,r(i)+2}=\cos\phi_i \tag{2c}$$

$$C^{r(i)}_{r(i)+1,r(i)+2}=-C^{r(i)}_{r(i)+2,r(i)+1}=\sin\phi_i \tag{2d}$$

with the cyclic convention for the index $r(i)$

$$r(i)+\ell\equiv r(i)+\ell-3 \qquad \text{if } r(i)+\ell>3 \tag{3}$$

If C denotes the transformation matrix from $\{a\}$ to $\{d\}$, then

$$C=C^{r(3)}(\phi_3)C^{r(2)}(\phi_2)C^{r(1)}(\phi_1) \tag{4}$$

and the inverse of C is given by[1,6]

$$C^{-1}=C^T=C^{r(1)}(-\phi_1)C^{r(2)}(-\phi_2)C^{r(3)}(-\phi_3) \tag{5}$$

The transformation matrix C of any sequence could be automatically generated on a digital computer by computing the elements of the matrices $C^{r(i)}(\phi_i)$ and carrying out the matrix multiplications of Eq. (4).[5] However, this simple algorithm is not appropriate for practical use, because this contains unnecessary multiplications such as $\cos\phi_1$ multiplied by zero or unity and so forth. With the help of basic matrix theory, such unnecessary multiplications are completely avoided by interchanging rows and columns in a systematic manner determined by the $r(j)$'s, as shown in the following (see Ref. 6 for detailed derivation of equations).

Let X be an arbitrary vector, then it may be expressed by the vector bases defined by Eqs. (1) and 3×1 matrices.

$$X=\{a\}^TX^a=\{b\}^TX^b=\{c\}^TX^c=\{d\}^TX^d$$

For the sake of programming convenience, the elements of X^a, X^b, etc. are represented by the subscripted variables, e.g.,

$$X^a=[X^a(1),X^a(2),X^a(3)]^T$$

It is noted that the numbers in the brackets of $X^a(\cdot)$, etc., refer to the axis number and not to the order of rotation. It turns out to be convenient to use the modular arithmetic function designated by MOD (i,j) for the cyclic convention of Eq. (3). If i and j are positive integers, then

$$\text{MOD}\ (i,j)=i-[i/j]\cdot j \tag{6}$$

where $[i/j]$ is an integer whose magnitude is the largest integer that does not exceed the magnitude of i/j and whose sign is the same as i/j.

Suppose that $r(i)$, ϕ_i $(i=1,2,3)$ and X^a are given and X^d is to be computed. Then, compute first the following indices:

$$\ell_1=r(1) \tag{7a}$$

$$\ell_2=\text{MOD}(\ell_1,3)+1 \tag{7b}$$

$$\ell_3=\text{MOD}(\ell_2,3)+1 \tag{7c}$$

$$m_1=\text{MOD}(r(2)-r(1)+3,3)+1 \tag{8a}$$

$$m_2=\text{MOD}(m_1,3)+1 \tag{8b}$$

$$m_3=\text{MOD}(m_2,3)+1 \tag{8c}$$

$$m_1'=\text{MOD}(r(3)-r(2)+3,3)+1 \tag{9a}$$

$$m_2'=\text{MOD}(m_1',3)+1 \tag{9b}$$

$$m_3'=\text{MOD}(m_2',3)+1 \tag{9c}$$

$$n_1=r(3) \tag{10a}$$

$$n_2=\text{MOD}(n_1,3)+1 \tag{10b}$$

$$n_3=\text{MOD}(n_2,3)+1 \tag{10c}$$

The computations of Eqs. (7)-(10) are required only once for a given order of $r(i)$, so that they are completely omitted as far as the $r(i)$ remain unchanged. Once these indices are obtained, the following computations need to be carried out every time when the ϕ_i change.

$$\hat{X}^b(1)=X^a(\ell_1) \tag{11a}$$

$$\hat{X}^b(2)=c_1X^a(\ell_2)+s_1X^a(\ell_3) \tag{11b}$$

$$\hat{X}^b(3)=-s_1X^a(\ell_2)+c_1X^a(\ell_3) \tag{11c}$$

$$\hat{X}^c(1)=\hat{X}^b(m_1) \tag{12a}$$

$$\hat{X}^c(2)=c_2\hat{X}^b(m_2)+s_2\hat{X}^b(m_3) \tag{12b}$$

$$\hat{X}^c(3)=-s_2\hat{X}^b(m_2)+c_2\hat{X}^b(m_3) \tag{12c}$$

$$X^d(n_1)=\hat{X}^c(m_1') \tag{13a}$$

$$X^d(n_2)=c_3\hat{X}^c(m_2')+s_3\hat{X}^c(m_3') \tag{13b}$$

$$X^d(n_3)=-s_3\hat{X}^c(m_2')+c_3\hat{X}^c(m_3') \tag{13c}$$

where $c_i=\cos\phi_i$ and $s_i=\sin\phi_i$ $(i=1,2,3)$, and $\hat{X}^b(\cdot)$ and $\hat{X}^c(\cdot)$ correspond to $X^b(\cdot)$ and $X^c(\cdot)$, respectively, with the elements interchanged in a specified manner. Eqs. (11)-(13) accomplish the transformation from X^a to X^d.

The transformation from X^d to X^a is similarly computed. The difference in computation of the indices is due to the order of the $r(i)$ accomplished by reordering them as $r(3)-r(2)-r(1)$. The calculations corresponding to Eqs. (11)-(13) need to be slightly modified in such a manner that the signs, plus or minus, of s_i $(i=1,2,3)$ are changed. Accordingly, the inverse transformation is computed in the same program with an appropriate indication of inversion.

Angular Velocities

The relation between the angular velocities, $\omega=[\omega_1, \omega_2, \omega_3]^T$ and the time derivatives of the attitude angles, $\dot{\phi}_1$, $\dot{\phi}_2$, $\dot{\phi}_3$, may be expressed in a general form[5,6]

$$\omega=[C^{r(3)}(\phi_3)\cdot C^{r(2)}(\phi_2)\delta^{r(1)}, C^{r(3)}(\phi_3)\delta^{r(2)}, \delta^{r(3)}]\begin{bmatrix}\dot{\phi}_1\\ \dot{\phi}_2\\ \dot{\phi}_3\end{bmatrix} \tag{14}$$

where $\delta^{r(i)}$ is a 3×1 matrix whose elements are the Kronecker delta and

$$\delta^{r(i)} = [\delta_1^{r(i)}, \delta_2^{r(i)}, \delta_3^{r(i)}]^T$$

Based on Eqs. (2) and (14), the angular velocity relation could be also computed on digital computer for any sequence of rotations, but as is pointed out previously, this simple algorithm includes unnecessary multiplications and additions. Such unnecessary calculations are shown to be avoided with the following simple algorithm.[6]

First, compute the indices, n_1, n_2, and n_3 (Eqs. (10)) and the following

$$k_1 = \mathrm{MOD}(r(2) - r(1) + 3, 3) \tag{15a}$$

$$k_2 = \mathrm{MOD}(r(3) - r(2) + 3, 3) \tag{15b}$$

$$k_3 = \mathrm{MOD}(k_2, 2) + 1 \tag{15c}$$

and

$$j_1 = 1 \quad \text{and} \quad j_2 = 2 \quad \text{if} \quad k_1 \neq k_2 \tag{16a}$$

$$j_1 = 2 \quad \text{and} \quad j_2 = 1 \quad \text{if} \quad k_1 = k_2 \tag{16b}$$

In addition to these, the following sign functions are defined:

$$p_1 = (-1)^{k_1+1}, \quad p_2 = (-1)^{k_2+1}, \quad p_3 = (-1)^{k_2} \tag{17}$$

Then, compute auxiliary functions $u(\bullet)$, $v(\bullet)$, etc., by

$$u(j_1) = \cos\phi_2 \tag{18a}$$

$$u(j_2) = p_1 \sin\phi_2 \tag{18b}$$

$$v(k_2) = \cos\phi_3 \tag{18c}$$

$$v(k_3) = \sin\phi_3 \tag{18d}$$

and

$$v_3 = p_3 v(2) \tag{18e}$$

$$v_4 = p_2 v(1) \tag{18f}$$

With this preparation, the elements of ω, designated by $\omega(i)$, $i = 1,2,3$ (which refer to the axis number) are given in terms of $\dot{\phi}_i$ $(i = 1,2,3)$

$$u_2 = u(2)\dot{\phi}_1 \tag{19a}$$

$$\omega(n_1) = u(1)\dot{\phi}_1 + \dot{\phi}_3 \tag{19b}$$

$$\omega(n_2) = v(1)u_2 + v(2)\dot{\phi}_2 \tag{19c}$$

$$\omega(n_3) = v_3 u_2 + v_4 \dot{\phi}_2 \tag{19d}$$

Conversely, $\dot{\phi}_i$ $(i = 1,2,3)$ are given in terms of ω

$$\dot{\phi}_1 = v(1)\omega(n_2) + v_3 \omega(n_3)/u(2) \tag{20a}$$

$$\dot{\phi}_2 = v(2)\omega(n_2) + v_4 \omega(n_3) \tag{20b}$$

$$\dot{\phi}_3 = \omega(n_1) - u(1)\dot{\phi}_1 \tag{20c}$$

As previously, the computations of Eqs. (15)-(17) are required only once for a given order of sequence, so that they are completely omitted as long as the $r(i)$ remain unchanged. Once these indices are computed, the computations of Eqs. (18) and (19) are to be computed for the case of $\dot{\phi} \rightarrow \omega$, or those of Eqs. (18) and (20) are for the case of $\omega \rightarrow \dot{\phi}$.

Conclusion

Algorithms have been established for computation of kinematical relations for three attitude angle systems used in digital computer simulation of spacecraft attitude dynamics. For this establishment, a general representation of a planar rotation is introduced (Eqs. (2)) and the transformation is accomplished as shown in Eqs. (7)-(13).

The established algorithms possess inversion capability, that is, with an appropriate indication of inversion the transformation from body to reference frame is computed in the transformation program from reference to body frame. In addition, computation of the time derivatives of three attitude angles from body rates, and the converse are carried out in the angular velocity computation algorithm [Eqs. (15)-(20)].

Both programs require as input the order of the rotation sequence and three attitude angles only, so that laborious manual calculation of matrices is completely avoided. Since particular care was paid to minimize the number of multiplications, there is no significant penalty on computation time in spite of the generality and utility of the programs.

References

[1] Likins, P.W., *Elements of Engineering Mechanics*, McGraw-Hill, New York, 1973, pp. 85-112.

[2] Pardoe, P.P., "A Description of the Digital Attitude Simulation," TG-964, Johns Hopkins University, Feb. 1968.

[3] Fang, A.C. and Zimmerman, B.G., "Digital Simulation of Rotational Kinematics," NASA TN D-5302, 1969.

[4] Mayer, A., "Rotations and Their Algebra," *SIAM Review*, Vol. 2, No. 2, 1960.

[5] Ohkami, Y., "Coordinate Transformation Algorithm by the Eulerian Angles Transformation," National Aerospace Lab., Tokyo, Japan, TR-305, Nov. 1972.

[6] Ohkami, Y., Computer Algorithms and Programs for Computation of Kinematical Relations for Three Attitude Angle System," National Aerospace Lab., Tokyo, Japan, to be published. NAL TR-305T.

Boundary-Layer Development on Moving Walls Using an Integral Theory

Kevin S. Fansler*
U.S. Army Ballistic Research Laboratories, Aberdeen Proving Ground, Md.
and
James E. Danberg†
University of Delaware, Newark, Del.

I. Introduction

THE aim of this work was to develop a technique which might be applied to compute the boundary-layer growth on moving walls. This method was applied to investigate the forces acting on a spinning cylinder in crossflow.[1]

The method presented here involves integrating the coupled integral-momentum and integral-energy equations for the dependent variables; a shape factor K and the momentum thickness in terms of the independent variable x, the distance along the wall. Other shape factors appearing in the integral equations are related to K and u_w/u_e (wall velocity to boun-

Received Nov. 3, 1975; revision received April 26, 1976.

Index category: Boundary Layers and Convective Heat Transfer—Laminar.

*Research Physicist, Exterior Ballistics Laboratory. Member AIAA.

†Professor of Mechanical and Aerospace Engineering. Associate Fellow AIAA.